

ADVANCING THE IMPLEMENTATION OF HYDROLOGIC MODELS
AS WEB-BASED APPLICATIONS

by

Prasanna Dahal

A thesis submitted in partial fulfillment
of the requirements for the degree

of

MASTER OF SCIENCE

in

Civil and Environmental Engineering

Approved:

David G. Tarboton
Major Professor

Jeffery Horsburgh
Committee Member

Joe Wheaton
Committee Member

Mark McLellan
Vice President for Research and Dean of
the School of Graduate Studies

UTAH STATE UNIVERSITY
Logan, Utah

2017

Copyright © Prasanna Dahal 2017

All Rights Reserved

ABSTRACT

Advancing the Implementation of Hydrologic Models
as Web-based Applications

by

Prasanna Dahal, Master of Science

Utah State University, 2017

Major Professor: Dr. David G. Tarboton
Department: Civil and Environmental Engineering

Advanced computer simulations are required to understand hydrologic phenomenon such as rainfall-runoff response, groundwater hydrology, snow hydrology, etc. Building a hydrologic model instance to simulate a watershed requires investment in data (diverse geospatial dataset such as terrain, soil) and computer resources, typically demands a wide skill set from the analyst, and the workflow involved is often difficult to reproduce. This work introduces a web-based prototype infrastructure in the form of a web application that provides researchers with easy to use access to complete hydrological modeling functionality. This includes creating the necessary geospatial and forcing data, preparing input files for a model by applying complex data preprocessing, running the model for a user defined watershed, and saving the results to a web repository. The open source Tethys Platform was used to develop the web app front-end Graphical User Interface (GUI). We used HydroDS, a web service that provides data preparation processing capability to support backend computations used by the app. Results are saved

in HydroShare, a hydrologic information system that supports the sharing of hydrologic data, model and analysis tools. The TOPographic Kinematic APproximation and Integration (TOPKAPI) model served as the driving use case for which we developed a complete hydrologic modeling service to demonstrate the approach. The final product is a complete modeling system accessible through the web to create input files, and run the TOPKAPI hydrologic model for a watershed of interest. An additional model, TOPNET, was incorporated to demonstrate the capability for adding other models into the framework.

(96 pages)

PUBLIC ABSTRACT

Advancing the Implementation of Hydrologic Models as Web-based Applications

Prasanna Dahal

Deeper understanding of relationship between flow in rivers and various hydrologic elements such as rainfall, land use, and soil type is imperative to solve water related problems like droughts and floods. Advanced computer models are essential in helping us understand such relationships. However, preparing such models requires huge investment of time and resources, much of which are concentrated on acquisition and curation of data. This work introduces a free and open source web application (web app) that provides researchers with easy to use access to hydrological data and modeling functionality. The web app helps in the creation of both hydrologic models, and climatic and geographic data. Free and open source platforms such as Tethys and HydroShare were used in the development of the web app. A physics based model called TOPographic Kinematic APproximation and Integration (TOPKAPI) model was used as the driving use case for which a complete hydrologic modeling service was developed to demonstrate the approach. The final product is a complete modeling system accessible through the web to create hydrologic data, and run a hydrologic model for a watershed of interest. An additional model, TOPNET, was incorporated to demonstrate the capability for adding other models into the framework.

ACKNOWLEDGMENTS

I would like to express my deepest gratitude to my major adviser, Dr. David Tarboton. Without his constant guidance, encouragement, and patience this work would not be possible. I also express my sincere gratitude for my former adviser, Dr. Anthony Castronova for not only mentoring me, but also for having belief in me to be his first student.

I am highly indebted to the Utah Water Research Laboratory (UWRL) for providing me with assistantship and funding for this work. Special thanks to Dr. Mac McKee and Dr. Jeffery Horsburgh for extending the support at, what could have been, a difficult time for me. Also thanks to Dr. Joe Wheaton for offering his valuable time to my committee.

This work would have been a difficult and unpleasant experience without the continuous encouragement and emotional support of my beloved wife Ruby Upreti. Special thanks to her for everything. Deepest thanks to my father Ramhari Dahal for his love. Thanks to all of my sisters for always taking good care of me.

Many thanks and appreciation to my colleagues Tian, Tsegeneh, Nazmus, Ayman, Adel and others for their kind suggestions and helps. Thanks to Awash Gautam (Midtone Graphics) for designing the app logo, and to Raman Upreti for his support. Also thanks to my Nepalese friends at Utah State who have made me feel home away from home.

Last but not least, this thesis has become a reality with the kind support of members from the stackoverflow community. I would like to extend my sincere thanks to all of them.

Prasanna Dahal

CONTENTS

	Page
ABSTRACT	iii
PUBLIC ABSTRACT	v
ACKNOWLEDGMENTS	vi
LIST OF TABLES	x
LIST OF FIGURES	xi
1. INTRODUCTION	2
1.1 Problem Statement.....	2
1.2 Research objectives and questions	3
2. BACKGROUND	5
2.1 Cyberinfrastructure.....	5
2.2 Hydrologic Cyberinfrastructure.....	6
2.3 HydroDS.....	8
2.4 HydroShare.....	9
2.5 Tethys Platform	10
2.6 Models	12
2.6.1 TOPKAPI.....	12
2.6.2 TOPNET	16

3. APP DESIGN AND DEVELOPMENT	18
3.1 Elements of the App	19
3.1.1 Methods Used to Access Data	19
3.1.2 Automate TOPKAPI Input-file Preparation	26
3.1.3 Retrieve Observed Flow	27
3.1.4 A System to Enhance Users' Access to the Computing Functionalities ..	27
3.1.5 GUI Based System for Inputs - Outputs	27
3.1.6 The App's Database.....	32
3.2 App Design.....	34
3.2.1 Overview of the App.....	34
3.2.2 Server Client Architecture	38
3.3 Experimental Use Cases	39
3.3.1 TOPKAPI Model	39
3.3.2 TOPNET Model.....	42
3.3.3 Download Hydrologic Dataset.....	43
3.3.4 Perform Global Terrain Analysis.....	44
4. RESULTS	45
4.1 Software Implementation	45
4.2 Use Case Results	47
4.2.1 TOPKAPI Model Results	47

4.2.2 Prepare Model-input Files for TOPNET.....	53
4.2.3 Download Hydrologic Dataset.....	54
4.2.4 Perform Global Terrain Analysis.....	56
5. DISCUSSION	58
6. CONCLUSIONS.....	64
REFERENCES	66
APPENDICES	71
Appendix A. Abbreviation and Glossary of Terms.....	72
Appendix B. SSURGO Dataset and Calculations	73
Appendix C. Terrain Analysis Using TauDEM	76
Appendix D. HydroDS Functions	78
Appendix E. Output JSON Summary File	80
Appendix F. App Tutorial: Prepare TOPKAPI Model.....	81
Appendix G. App Tutorial: Prepare TOPNET Input Files.....	89
Appendix H. App Tutorial: Perform Terrain Analysis.....	92
Appendix I. Source Codes Locations	96

LIST OF TABLES

Table	Page
1. Brief description of the files created during the terrain analysis.....	20
2. Lookup table that maps NLCD land cover classes to Manning's n values.....	21
3. Brief description of the files created during preparation of soil dataset.....	23
4. Lookup table for soil properties based on soil texture class	23
5. Brief description of the files created during preparation of forcing dataset.....	24
6. The input parameters used for creating the TOPKAPI model.....	41
7. The input parameters used for creating input files for TOPNET model	42
8. The input parameters used in downloading hydrologic dataset	42
9. A list of information given for performing terrain analysis	43
10. A list of hydrologic dataset created.....	54
11. A list of terrain dataset created.....	56

LIST OF FIGURES

Figure	Page
1. Schematic diagram of architecture of HydroDS data services	9
2. Schematic diagram of how HydroShare works	10
3. Schematic diagram of Tethys Software Platform.....	11
4. Flowchart showing Penman-Monteith based calculations executed during creation of gridded short crop reference evapotranspiration datasets.....	26
5. The two major sections in the home page.....	29
6. The two major sections in the model run page.....	31
7. The web request architecture implemented in the app	32
8. Datamodel used in the app	33
9. The processes carried out and the inputs used during the execution of a model preparation request by a user	36
10. Diagram showing the relationship between the app, HydroDS and HydroShare..	38
11. Study site location and DEM of Santa Cruz watershed	40
12. Model input page of the app.....	45
13. Model run page of the app.....	46
14. Result provided after the model run: simulated and observed hydrograph	48
15. Result provided after the model run: Time-series plots	48
16. Result provided after the model run: Cumulative volume of water entering and leaving the watershed	49
17. Initial parameters value used while creating the model	49
18. Results obtained after rerunning the model with a changed parameters value....	51

19.	Changed initial parameters value used to modify the model	51
20.	Screenshot of the links to the results displayed by the app after TOPNET input-file preparation	54
21	Some of the hydrologic dataset created for the experimental watershed.....	55
22.	Some of the terrain dataset created for the Bagmati River at Kathmandu.....	56

1. INTRODUCTION

1.1 Problem Statement

This thesis aims to address the time consuming and repetitive input file preparation related problems that exist during the preparation of a hydrologic model. Particularly, the problems of data discovery, retrieval, Geographic Information System (GIS) preprocessing and other data analyses are addressed.

Development of hydrologic model instances are necessary because advanced computer simulations are required to understand hydrologic phenomenon such as rainfall-runoff response, groundwater hydrology, snow hydrology, etc. (Singh and Frevert, 2002). However, building a hydrologic model to simulate a watershed requires investment in data (diverse geospatial dataset such as terrain, soil) and computer resources, and typically demands a wide skill set from the analyst. A typical hydrologic model setup involves a labor-intensive data preparation phase with workflows that may be difficult to reproduce (Ames et al., 2012; Ames et al., 2015; White, 2012; Taylor et al., 1999; Granell et al., 2010). Building a hydrologic model requires skills such as accessing data from different sources and performing complex preprocessing actions on the data often using expensive GIS tools and programming scripts (Taylor et al., 1999; Granell et al., 2010). Therefore, the data aspects of hydrologic models have often been a barrier to their widespread use in solving watershed problems (Choi et al., 2005).

Model and model input preparation problems also include machine dependencies, which limit a set of code or software to only working in a particular machine or environment. For example, software such as ArcGIS Desktop is only available on Windows, so Macintosh or Linux users have challenges in using such software. A

hydrologist is often required to invest considerable time to learn and implement the skills required during pre- or post-processing of geospatial data- without the guarantee of them being useful in the all platforms.

Therefore, data and model platform aspects of hydrologic models have often been a barrier to their widespread use in solving watershed problems (Choi et al., 2005). Automated computer-based procedures to store, access, and prepare data for modeling are becoming essential to tackle these barriers (Miller et al., 2007). The goal of my research was to investigate the development of a prototype system to reduce complexity involved in the application of a hydrologic model.

1.2 Research objectives and questions

Granell et al., (2010) and Goodall et al. (2008) have noted that much hydrologic analysis and modeling is constrained by time-consuming data processing and management tasks. Acquiring, organizing and reformatting data for use in study often demands a wide skill-set from a hydrologist that is not directly related to hydrology. So, a question is- what is the best use of modern computer technology to make it easier for a hydrologist to prepare hydrologic models quickly and easily, without the hassle of installing software, and with an ability to reproduce results? The answer may be found by observing the increasing trend towards web-hosted services for watershed management and GIS (Choi et al., 2005; Pandey et al., 2000; McKee, 2000; Granell et al., 2010; Mineter et al., 2003). Web-based solutions offer significant advantages over more traditional approaches because users are not required to own and maintain specialized, and sometimes platform dependent software. Further, web based solutions can provide intuitive GUIs (Graphical User Interface) for input and output management, and intensive data processing can be accomplished on

powerful servers (Pandey et al., 2000). For these reasons, a web application (web-app) was investigated in this research as a way to address the problems described.

The specific objective of the study were: i) evaluate the development of a web app to provide a user-friendly system to create input files for the TOPKAPI model rapidly, with high accuracy, without going through the tedious process of downloading, processing, and analyzing data on the user's machine; ii) evaluate web based model execution of the TOPKAPI model instance using input files from the web app; iii) evaluate providing users the ability to create geospatial and forcing files like terrain rasters, soil properties rasters, rainfall and reference evapotranspiration (ET) for a modeling area of interest, and iv) investigate the leveraging of these functionalities to developers programmatically through Application Program Interface (APIs).

2. BACKGROUND

This section gives general background on cyberinfrastructure for hydrologic modeling and research in general. The first section introduces the concept of Cyberinfrastructure. Then in the second section general hydrologic cyberinfrastructure work is reviewed. The next three sections review each of the components used in the web-app that I developed, namely: i) the Tethys Platform which was used to build the web interface, ii) the HydroDS web service which served as the engine of the app, and iii) HydroShare which was used to save the result files created by the app. The last subsection introduces the models supported by the app: TOPKAPI, and TOPNET.

2.1 Cyberinfrastructure

The National Science Foundation (NSF) started a comprehensive program called the Cyberinfrastructure (CI) Program to advance the cyberinfrastructure in all areas of science and engineering research and allied education (Atkins, 2003). The program supports the idea of adding ‘computer simulations’ as a major approach to the two classic scientific research approaches: theoretical/analytics and experimental/observational research. The advancement of CI is crucial in achieving the vision of ‘web science’ mentioned earlier by giving researchers easy access to the processes, tools, data representation, architecture, infrastructures, and computationally-intensive modeling to help facilitate easy discovery, processing, integration, analysis of hydrologic data from disparate data sources. This research carries the theme of contributing to the improvements of cyberinfrastructure.

2.2 Hydrologic Cyberinfrastructure

Right from the rise of personal computers (PC), people have imagined the use of PCs in interactive and easy-to-use simulation services for better understanding of environmental resources allowing more informed decision making for environmental planning and policy making (Loucks et al., 1985). Many applications exist that have offered users services of environmental modeling, or model input preparation. Although their architectures differ quite sharply, they could be broadly classified into two major groups: PC-based applications, and web-based applications. PC-based applications (Best et al., 2007; Teng et al., 2008; Pecar-Ilic and Ruzic, 2006; Mineter et al., 2003; Jeong et al., 2006; Miles and Band, 2015; Miller et al., 2007) either work as standalone programs built for a particular operating system, or work inside the environment of another program (e.g. ArcGIS) as a tool or a plugin. These applications are designed for a host PC, hence they are machine dependent. Furthermore, access to such applications can be limited because of the licensing restrictions, dependencies issues and software versioning.

On the other hand, web-based applications offering modeling and input-preparation services (Goodall et al., 2008; Soh et al., 2006; Leonard and Duffy, 2013; Dawson et al., 2007; Walker and Chapra, 2014; Lim and Engel, 2000; Granell et al., 2010; Choi et al., 2005) do so by applying data preprocessing and GIS transformation in web hosted machines. Users access these applications via the web, while the computations are carried out in the host server. Users do not run any application in their local machines; hence, these applications are not subjected to machine dependency. However, they may require accounts to use and their persistence, longevity and stability is at the mercy of the host organization. A user has control over when or whether to update PC based software, while

a website may be updated or go offline without warning.

Web-based applications providing hydrologic model input preparation include the Essential Terrestrial Variable (ETV) web service (Leonard and Duffy, 2013) that offers generic hydrologic dataset preparation for watershed modeling for United States Geological Survey (USGS) Hydrologic Unit Code (HUC) units, and a system to transform the ETV into Penn State Integrated Hydrological Model (PIHM) input file formats. Also, Granell et al. (2010) used geospatial services for discovery, access, processing and visualization of geospatial data in a distributed manner applied to two runoff models- Snowmelt Runoff Model and TUW-HBV model. Similarly, a team from Purdue University has developed an online system using web-GIS that performs online watershed delineation, data preparation and online simulation using web-accessible version of Long-Term Hydrological Impact Assessment L-THIA (Choi et al., 2005) system to model the impact of land-use change scenarios on runoff and nonpoint source pollutant. These works collectively illustrate the computing capabilities of each of these applications in terms of processing, reformatting, analyzing, transforming geospatial dataset, and carrying out model specific computations. However, there are still opportunities to improve and extend systems for web based model input preparation and model execution.

There have also been studies to make hydrologic data from disparate sources easily available. The GeospatialDataGateway has been a reliable public servant offering different geospatial datasets to users easily through its website <https://datagateway.nrcs.usda.gov/>. PC based software such as HydroDesktop (Ames et al., 2012) gives users ability to interactively discover and download different types of hydrologic data from many sources. The Consortium of Universities for the Advancement of Hydrologic Science, Inc.'s

Hydrologic Information System (CUASHI- HIS) has offered a web service that follows open standard for the exchange of water observation data called Water Markup Language (WaterML) for programmatically accessing and connecting between data from government repository and academic hydrologic observation network (Horsburgh et al., 2009; Zaslavsky et al., 2007). Although such software systems are useful and popular because they act as one stop solution for data discovery, they do not focus on preparing inputs for any specific models, which means a lot of preprocessing could still be required to use these dataset in environmental models.

The work described above contributes and builds towards broader visions such as Web Science (Berners-Lee et al., 2006; Shneiderman, 2007; Granell et al., 2010), and Model as a Service (Geller and Melton, 2008; Geller and Turner, 2007; Roman et al., 2009). Such visions foresee the integration of web related capabilities like data discovery, processing and analyzing datasets from disparate data sources, and accessing different models provided as web services. With the implementation of modeling services as a web-based application, this project too aims to be a step towards achievement of the broader vision too.

2.3 HydroDS

Sazib (2016) describes a web service called HydroDS (Figure 1) which supports data and GIS processing services such as downloading geospatial and climate data and raster manipulation for preparation of model inputs through a REST API (Representational state transfer API). It also supports some modeling capabilities. A python client library, called *Hydrogate* (Dash, 2016) provides functions that call the HydroDS API from Python. This library makes it easy to use these web services from a program written in Python.

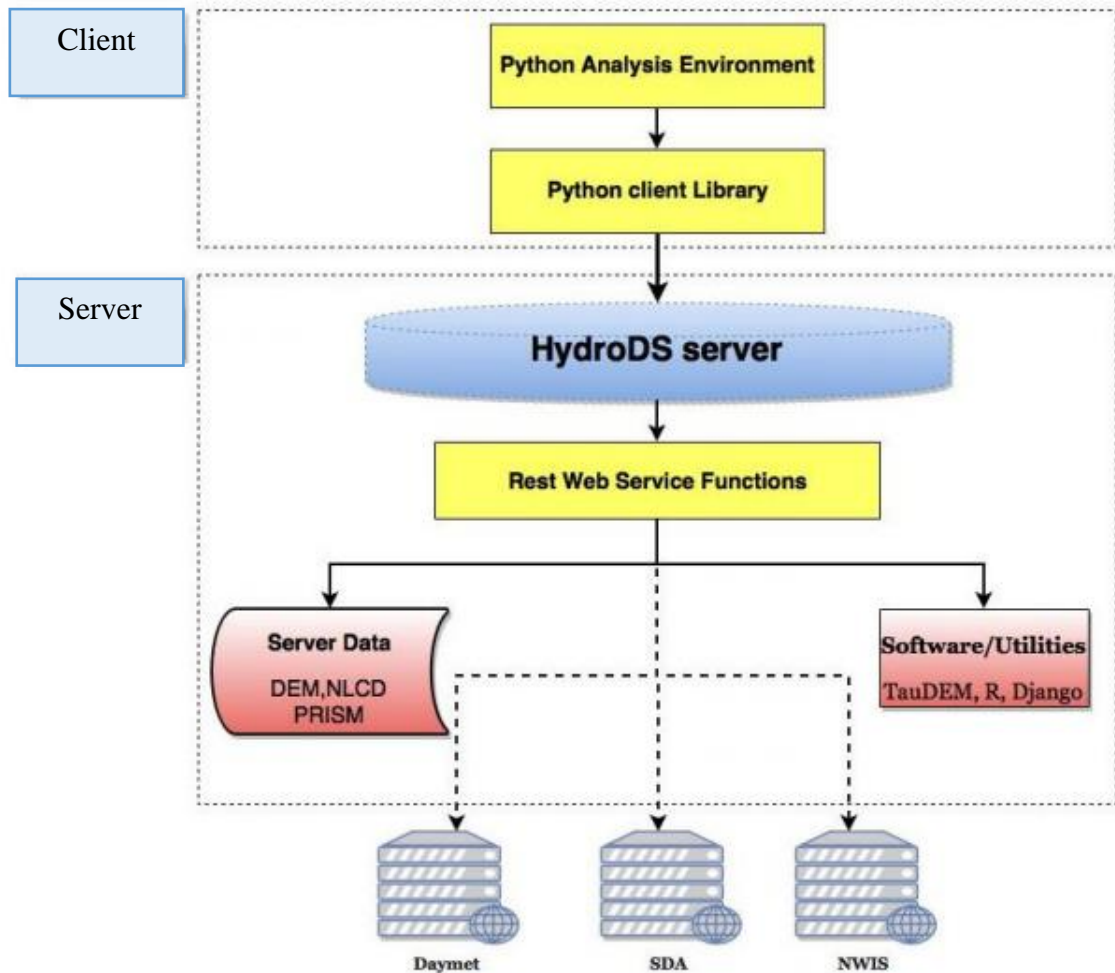


Figure 1. Schematic diagram of architecture of HydroDS data services (Sazib, 2016). The HydroDS server contains data (DEM, NLCD etc.), as well as packages such as TauDEM and R. The client element in our case is Tethys backend.

2.4 HydroShare

HydroShare is an online system developed for collaborative sharing of hydrologic data and models with the aim to enable scientists to easily discover and access hydrologic data and models, and perform analyses on them in a distributed computing environment (Tarboton et al., 2013). HydroShare is comprised of three primary components (Tarboton

et al., 2014): (1) Distributed file storage in Integrated Rule-Oriented Data System (iRODS) (Rajasekar et al., 2010); (2) a website for exploring and interacting with content comprising data and models, collectively referred to as resources; and (3) Web applications that are tools and programs that can act on resources (Figure 2). HydroShare is extensible in that anyone can develop a web application that interacts with HydroShare resources through its various APIs. The app makes use of HydroShare REST API Python client library called `hs_restclient` (<http://hs-restclient.readthedocs.io/>) to communicate with HydroShare.

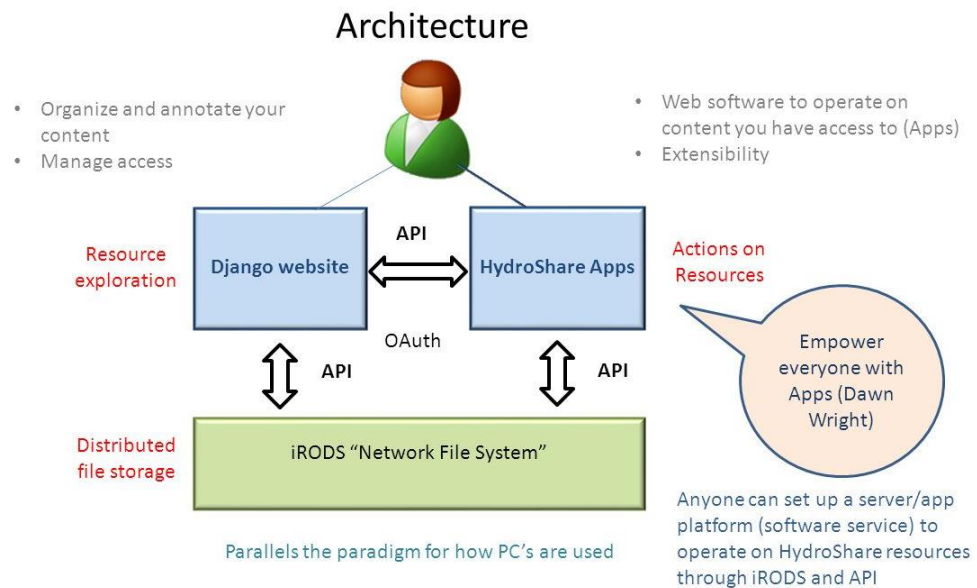


Figure 2. Schematic diagram of how HydroShare works (Tarboton, 2016)

2.5 Tethys Platform

The Tethys Platform (Swain et al., 2016) was developed to make it easy to build web apps for water resources applications. Tethys is a Django based development and hosting environment developed using a Python Software development kit (SDK) which aims to lower the barrier for web app development (Swain et al., 2016; Ames et al., 2015) by bringing together several Free and Open Source (FOSS) components together.

A conceptual diagram and overview of the platform as described by Swain (2016) is shown in Figure 3 below. End users can access an app built using the Tethys Platform on their PCs or handheld devices. Tethys platform allows the use of map elements such as Google earth, Google maps, OpenLayers, “gizmos” elements (pre-programmed interactive controls) such as date picker, text inputs, buttons, sliders, and HighCharts tool for visualization of time series and tabular data. It also allows the use of PostGIS spatial database.

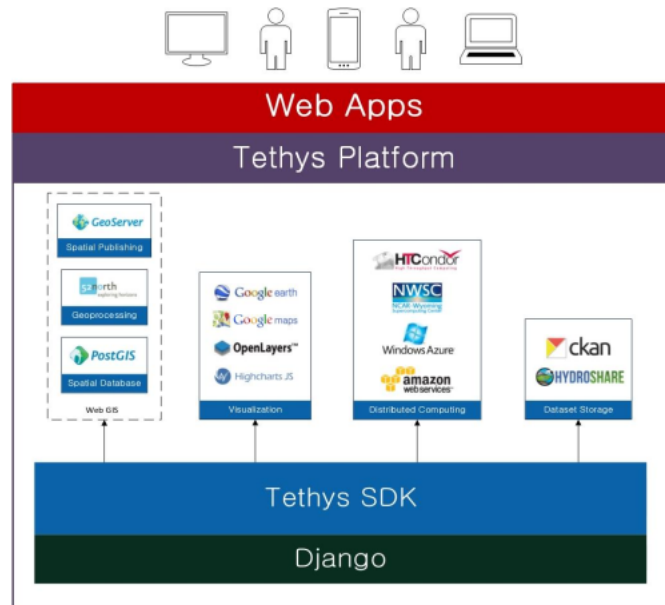


Figure 3. Schematic diagram of Tethys Software Platform (Swain, 2016)

Tethys apps use a development pattern called “Model View Controller”. Swain (2015) defines this as “the Model represents the data of the app, the View is composed of the representation of the data, and the Controller consists of the logic needed to prepare the data from the Model for the View and any other logic your app needs”.

The controller coordinates between the view and the model. The Controller handles most of the application logic such as processing form data, launching model runs, querying database or calling specific functions. In a Tethys app, controllers are python functions.

2.6 Models

This section introduces the two models used in the app- TOPKAPI and TOPNET. Both TOPKAPI and TOPNET are examples of physically based hydrologic models. Physically based distributed hydrologic models can, in principle, better model an ungauged catchment through their use of parameters which have a physical interpretation and through their representation of spatial variability in the parameter values (Abbott et al., 1986; Beven, 1989; Beven et al., 1984). This is the main reason for selection of the two model in the app.

2.6.1 TOPKAPI

The TOPKAPI (TOPographic Kinematic APproximation and Integration) model is a fully distributed physically-based hydrologic model that is derived from the integration in space of the kinematic wave model. The TOPKAPI model was evaluated as part of National Oceanic and Atmospheric Administration (NOAA's) Distributed Model Inter-comparison Project Phase 2 (DMIP2), where promising preliminary results have been reported (Coccia et al., 2009). The TOPKAPI model has been applied to several catchments for flood forecasting, extreme flood analysis, predicting hydrological response under changed landscape conditions (Liu and Todini, 2002; Todini and Ciarapica, 2001). Some of the countries in which TOPKAPI has been used include China (Liu et al., 2005), Italy

(Ciarapica and Todini, 2002; Bartholmes and Todini, 2005), South Africa (Sinclair and Pegram, 2010; Vischel et al., 2008b), and Nepal (Shea et al., 2015; Pellicciotti et al., 2012).

Following is the description of the model based on the text by Todini and Ciarapica (2002) and Coccia et al., (2009) paper. Readers are encouraged to review these references for more detailed understanding of model components and process parameterizations.

The TOPKAPI model combines the kinematic approach with the basin topography described by a lattice of square cells that are computational nodes. Cell size often increases with the scale of the problem. The flow paths are derived from a DEM with any one cell draining to one adjacent, but not diagonal neighboring cell while it may receive upstream contribution from up to three adjacent surrounding cells. Each cell is structured to represent soil component, overland flow component and drainage network component, with no component accounting for water percolation to deeper soil layers. The horizontal flow in the unsaturated zone is assumed to occur in a layer of limited thickness. Overland flow is generated by the excess rainfall on the saturated cells while the total runoff (surface + subsurface) is then drained by the drainage network.

The basic assumptions upon which the soil model for TOPKAPI is based are:

- i) The precipitation is assumed to be constant over a cell.
- ii) All the precipitation falling on the soil infiltrates, unless the soil has reached saturation
- iii) The slope of the water table coincides with the slope of the ground, unless the ground slope is very small (less than a minimum specified in a parameter, e.g. 0.1% in which case slope is taken as the minimum). This constitutes the fundamental assumption in the kinematic wave

approximation, and it implies the adoption of a kinematic wave propagation model for horizontal flow, in the unsaturated area.

- iv) The saturated hydraulic conductivity is constant with the depth in a surface soil layer but much larger than that of deeper soil layers which are neglected in the model representation of subsurface flow processes.
- v) The local transmissivity (integral of the hydraulic conductivity over the vertical in the unsaturated zone) can be reasonably expressed as a function of the total water content of the soil.

Transmissivity for the unsaturated soil layer may be defined as:

$$T = \int_0^L k(\tilde{\nu}(z)) dz \quad (1)$$

where L is the thickness of the layer affected by the horizontal flow, z is the vertical direction, $k(\tilde{\nu}(z)) = k_s \cdot (\tilde{\nu}(z))^\alpha$ is the hydraulic conductivity given by Brooks and Corey (1964), for non-saturated condition, as a function of the reduced water content $\tilde{\nu} = \frac{\vartheta - \vartheta_r}{\vartheta_s - \vartheta_r}$.

According to assumption iv) and v), the transmissivity T defined above does not strongly differ from the one estimated in terms of total soil moisture content integrated along the vertical profile (Benning, 1994):

$$T(\tilde{\Theta}) = k_s L \tilde{\Theta}^\alpha \quad (2)$$

where k_s is the saturated hydraulic conductivity, $\tilde{\Theta} = \frac{1}{L} \int_0^L (\tilde{\nu}(z)) dz$ is the mean value of reduced soil moisture content along the vertical profile, and α is a Brooks and Corey parameter which depends on soil characteristics.

Thus, while the horizontal flux calculated for the soil profile by Brooks and Corey formula is:

(3)

$$q = \int_0^L \tan(\beta) k_s \cdot (\check{v}(z))^\alpha dz$$

the horizontal flux calculated for each cell by means of the approximated formula is:

$$q = \tan(\beta) k_s L \check{\Theta}^\alpha \quad (4)$$

The model of the single cell can be derived from the continuity of mass and an approximated momentum equation, expressed after the vertical lumping as:

$$(\vartheta_s - \vartheta_r) L \frac{\partial \check{\Theta}}{\partial t} + \frac{\partial q}{\partial x} = r \quad (5)$$

where:

- x: cell size (horizontal)
- t: time coordinate
- q: horizontal flow in soil as discharge per unit width
- r: rainfall intensity

The equations (4) and (5) are combined and rewritten in terms of total soil moisture content over the vertical profile:

$$\eta = (\vartheta_s - \vartheta_r) L \check{\Theta} \quad (6)$$

to give:

$$\frac{\partial \eta}{\partial t} = r - \frac{\partial q}{\partial x} \quad (7a)$$

$$q = \frac{\tan(\beta) k_s L}{(\vartheta_s - \vartheta_r)^\alpha L^\alpha} \eta^\alpha = C \eta^\alpha \quad (7b)$$

where horizontal flow is expressed as a function of a local conveyance

$$C = \frac{\tan(\beta) k_s L}{(\vartheta_s - \vartheta_r)^\alpha L^\alpha} \quad (8)$$

which depends on physically meaningful soil properties. Combining the two equations (7)

the following kinematic equation is finally obtained:

$$\frac{\partial \eta}{\partial t} = r - \frac{\partial q}{\partial x} = \frac{\partial (C \eta^\alpha)}{\partial x} \quad (9)$$

This can be integrated in the soil over the i^{th} cell to give:

$$\frac{\partial V_{si}}{\partial t} = rx - (C \eta_{si}^{\alpha} - C_{si-1} \eta_{si-1}^{\alpha_s}) \quad (10)$$

where V_{si} is the volume per unit width stored in i^{th} cell, while the last term in equation (10) represents inflow and outflow balance. The subscript s is added to indicate that this equation is for soil cells.

The app introduced by this project uses PyTOPKAPI (Vischel et al., 2008b; Vischel et al., 2008a; Sinclair and Pegram, 2010; Sinclair and Pegram, 2013), an open-source implementation of the physically based distributed model TOPKAPI written in Python. It uses an upgraded TOPKAPI model that supports an active cell receiving flow from up to 7 surrounding cells, and allows flow along diagonals between cells. It also includes Green Ampt infiltration as a mechanism of overland flow generation. However, it does not have a snow modeling component.

2.6.2 TOPNET

TOPNET applies TOPMODEL to multiple sub-basins (as basic model elements) draining to distinct segments of a stream network. The model applied to each sub-basin is a modification of the original TOPMODEL (Beven and Kirkby, 1979) that adds a potential evapotranspiration component, a canopy storage component to model interception, and a soil zone component that provides infiltration excess runoff generation capability through a Green-Ampt like parameterization and other modifications (Bandaragoda et al., 2004).

For each sub-basin model, there is a separate interception component that controls the production of net precipitation. Overland flow and seepage from the groundwater form the inflow to each channel segment. The sub-basins are linked by a network of channels

and flow is routed through this network using kinematic waves with a shock fitting technique. It provides a prediction of flow in each modelled reach within a catchment (Bandaragoda et al., 2004; Clark et al., 2008).

TOPNET has been used in flow prediction, flood forecasting, water resources modelling, and climate and land-use change studies (Sazib, 2016; McMillan et al., 2010; Ibbitt and Woods, 2004).

3. APP DESIGN AND DEVELOPMENT

In order to lower the data related barrier present during the use of a hydrologic model, a web app was developed. The web app consists of:

1. Methods to access hydrological data required for modeling.
2. A set of programming scripts to automate hydrologic model preparation and modification of existing model instances.
3. A set of programming scripts to retrieve observed flow, and calculate error of simulated flow based on the observed flow.
4. A GUI based framework with html forms and draw-able map element to receive input parameters from a user, display time-series results of a model run, and allow loading of existing model, as well as its modifications.
5. A system to enhance users' access to the individual hydrologic functions used in modeling and model input preparation.
6. A database to store model input information.

Each element will be described in detail in the next section, “Elements of the App” that includes subsections for each element, with additional detail in the appendices. Then there is a section ‘App Design’ that describes the design used to bring together these concepts to make an app that can perform beginning to end hydrologic modeling. Finally we describe experimental use cases that were used to evaluate and validate the functioning of the app.

3.1 Elements of the App

3.1.1 Methods Used to Access Data

Python functions have been written to access a number of different hydrological datasets required for hydrologic modeling. Various GIS preprocessing tasks are used to convert data to the format required by the model. Some of the required data are stored on the HydroDS servers while others are fetched dynamically from various web services. The data are grouped into three types- terrain data, soil data, and forcing data. The procedures followed for each of these datasets are described in the sections below:

Terrain Dataset

30 m resolution National Elevation Dataset-Digital Elevation Model (NED-DEM), is stored locally in the HydroDS server for the Western United States, while 30 m resolution National Land Cover Dataset 2011 (NLCD) is stored for the contiguous United States (CONUS). For areas outside CONUS, Shuttle Radar Topography Mission (SRTM) DEM (30 meter resolution) is downloaded on the fly using a web service accessed from a python package named Elevation (<https://pypi.python.org/pypi/elevation>). Programming scripts were written to extract the dataset for the specified model domain and to resample and project it to the specified resolution.

TauDEM (Tarboton, 2005) tools are used to perform terrain analysis tasks such as delineate stream network, identify terrain driven surface flowpath, and watershed area draining into an outlet, a point of interest to which water falling on the entire watershed converges. Pits in the DEM for the region are filled. Then, the stream network is obtained from the DEM based on the user input “stream threshold” value required to define a

channel. Stream threshold value is total number of cells required to drain to a cell to be considered as a stream cell. The user input outlet information may not be accurate, i.e. the outlet may not fall exactly on the stream. For such cases, TauDEM function ‘moveoutlettostream’ is used to create a new shapefile containing a point for outlet in the stream. The final output of the analysis include a DEM, flow direction map, stream network map, slope map, watershed coverage map. The workflow carried out for the analysis, along with the TauDEM functions used to do so, is shown in Appendix C Figure C.1. Table 1 lists the data produced during the terrain analysis steps.

Table 1. Brief description of the files created during the terrain analysis

Filename	Descriptions
DEM84.tif	NED DEM (or SRTM DEM) clipped to the user domain, in WGS84 Coordinate System.
DEM84_prjXX.tif	The DEM in a projected coordinate system
fel.tif	TauDEM produced pit-filled DEM
src.tif	TauDEM produced stream network raster
p.tif	TauDEM produced d-8 flow direction raster
sd8.tif	TauDEM produced d-8 slope raster
ad8.tif	TauDEM produced d-8 area contributing raster
slope.tif	TauDEM produced d-8 tan(slope)
mask.tif	Watershed raster that drains to the outlet provided
mask_wgs.tif	Watershed raster in WGS84 coordinate system
nlcdProj.tif	Projected NLCD raster
Outlet.shp (.shx, .dbf, .prj)	ESRI point shapefile created based on location of outlet provided by the user
OutletProj.shp	The outlet shapefile in projected coordinate system
Corrected_outlet.shp	The outlet shapefile that falls on the stream network
watershed.shp	ESRI polygon shapefile for the watershed
metadata.txt	Text file containing metadata, including a dictionary of the details provided during the terrain analysis

NLCD for the domain is used to determine Manning’s n for the region. Kalyanapu et al. (2010) has mapped each NLCD land cover class to a Manning’s n value (Table 2). Using that table, a GIS reclassify function is used to determine a Manning’s n value for overland flow cells.

Table 2. Lookup table that maps NLCD land cover classes to Manning's n values (from Kalyanapu et al., 2010)

Land Cover Class	Description	Manning's n
21	Developed, open space	0.0404
22	Developed, low intensity	0.0678
23	Developed, medium intensity	0.0678
24	Developed, high intensity	0.04044
31	Barren land	0.0113
41	Deciduous forest	0.36
42	Evergreen forest	0.32
42	Mixed forest	0.4
52	Shrub/scrub	0.4
71	Grassland/herbaceous	0.368
81	Pasture/Hay	0.325
90	Woody wetlands	0.086
95	Emergent herbaceous wetlands	0.1825

It is to be noted that PyTOPKAPI requires the cells to be in equal area projected coordinate system. For the simplicity of accommodating any watershed in the Western US, we have adopted the North America Albers Equal Area Conic projected coordinate system (EPSG Code: 102003) for all the terrain datasets.

Soil Dataset

A set of programming scripts were written to prepare soil properties maps from Soil Survey Geographic Database (SSURGO) for a specified spatial domain. These include saturated hydraulic conductivity, porosity, bubbling pressure, residual soil moisture content, saturated soil moisture content, soil depth, and pore-size distribution (Table 3). A public soil database named gSSURGO (gridded SSURGO) provided by United States Department of Agriculture- Natural Resources Conservation Service (USDA-NRCS),

which is available throughout the United States, is used to extract soil data for the domain. SSURGO is a relational database containing tables describing properties, and maps. The maps are linked in the database to information about the ‘component soils’ and their properties for each ‘map unit’. Each component is further divided into ‘soil horizon’. During the process of obtaining a soil map for an area, often weighted averaging of values from soil horizon, and later from map components was carried out based on the description provided by (Wieczorek, 2014).

Offline storage of soil database for the entire US, or even the Western US is extremely difficult because of huge storage requirements. Hence, an R package named ‘soilDB’ (Beaudette and J. Skovlin, 2016) is used, which makes use of NRCS web services to query the database hosted in an NRCS server, and return the results. The results returned is used to construct a raster for different soil properties.

Table 3. Files created during preparation of soil dataset

Filename	Descriptions
Soil_mukey.tif	A raster containing SSURGO map unit keys
SSM.tif	Saturated Soil Moisture content (unitless) raster prepared by weighted averaging the soil property value (in this case saturated soil moisture values derived for each soil texture class from Table 2) for different soil horizons based on horizon’s thickness, and the weighted averaging them based on component percentage to map units..
RSM.tif	Residual Soil Moisture content raster (unitless) prepared by weighted averaging algorithm as explained before
BBL.tif	Bubbling pressure (in mm) raster prepared based on weighted averaging algorithm as explained before
PSD.tif	Pore size distribution raster (unitless) prepared based on weighted averaging algorithm as explained before
ksat_ssurgo_wtd.tif	Saturated soil conductivity raster (m/hr) prepared based on weighted averaging of values by SSURGO
ksat_LUT.tif	Saturated soil conductivity raster (mm/s) prepared based on weighted averaging algorithm as explained before

mannings_n.tif	Manning's n raster for the overland, derived from mapping NLCD raster
psif.tif	Green Ampt wetting front suction in meters
dth1.tif	Drainable porosity
dth2.tif	Plant available porosity
texture_joint_df.csv	A comma separated file containing dataframe of soil properties for each soil horizon from SSURGO for that has been joint to the look up table
component_agg_df.csv	A comma separated file containing weighted averaged soil properties for each soil component
mapunit_agg_df.csv	A comma separated file containing dataframe of the soil properties for each map unit keys
metadata.txt	Text file containing metadata, including a dictionary of the details provided during the terrain analysis

Multiple queries to NRCS are made with the help of the R package 'soilDB'; beginning with a query to get the shape information for mapunits in vector format. Then, another query is made to download a table of horizon level data from SSURGO. The table is weighted averaged (first based on thickness of soil horizon layer, and then based on percentage of different soil components) following the steps provided by Wieczorek (2007) to get values such as saturated hydraulic conductivity and porosity, for each map unit. Other soil properties are not directly available in the database, hence a lookup table based on soil texture class (Table 4) provided by Rawls (1982) was used to obtain properties such as saturated soil moisture content and bubbling pressure for each map unit. Missing values are replaced with the average value for the watershed. Further explanation is provided in Appendix B.

Table 4. Lookup table (Rawls, 1982) for soil properties based on soil texture class

Texture class	Porosity (cm ³ /cm ³)	Residual Soil Moisture (cm ³ /cm ³)	Bubbling pressure (mm)	Pore Size Distribution	Saturated Hydraulic Conductivity (mm/s)
Sand	0.437	0.02	72.6	0.592	0.05833
Loamy sand	0.437	0.035	86.9	0.474	0.01697
Sandy loam	0.453	0.041	146.6	0.322	0.00719

Loam	0.463	0.027	111.5	0.22	0.00367
Silt loam	0.501	0.015	207.6	0.211	0.00189
Sandy clay loam	0.398	0.068	280.8	0.25	0.00119
Clay loam	0.464	0.075	258.9	0.194	0.000639
Silty clay loam	0.471	0.04	325.6	0.151	0.000417
Sandy clay	0.43	0.109	291.7	0.168	0.00033
Silty clay	0.479	0.056	341.9	0.127	0.00025
Clay	0.475	0.09	373	0.131	0.000167

Forcing Dataset

Daymet climate files (precipitation, daily maximum and minimum temperature, short wave radiation, vapor pressure etc.) in a three dimensional NetCDF (Network Common Data Form) format for the entire US have been stored in the HydroDS server for the period ranging from 2005-2015. The app allows user access to these climate and forcing dataset (Table 5) for their area of interest in a gridded form.

Precipitation records for the user's domain and time range are extracted and resampled to the resolution of other geospatial files.

Table 5. Brief description of the files created during preparation of forcing dataset

Filename	Descriptions
Rain.nc	A 3-dimensional (X,Y and time) NetCDF file containing Daymet precipitation that has been resampled to user desired resolution (mm/day)
ET_reference.nc	A 3D NetCDF file containing short crop reference ET calculated using Penman-Monteith equations, and using Daymet climate dataset (mm/day)
Output_tmax.nc	A 3D NetCDF file of maximum daily temperature (°c)
Output_tmin.nc	A 3D NetCDF file of minimum daily temperature (°c)
Output_vp.nc	A 3D NetCDF file of vapor pressure (Pascal)
Output_srad.nc	A 3D NetCDF file of short wave radiation (W/m ²)
metadata.txt	Text file containing metadata, including a dictionary of the details provided during the creation of forcing files

A set of programming scripts was written to calculate reference ET based on Daymet data. The scripts use the DEM and the Daymet climate files- maximum temperature, minimum temperature, short wave radiation and vapor pressure. The calculations were based on the Food and Agriculture Organization (FAO) Penman-Monteith equations and guidelines (Allen et al., 2006).

The workflow followed during the creation of the ET product is shown in Figure 4. Total daily incoming solar radiation (in MJ/m²/day) for the region is calculated from Daymet short wave radiation dataset (in W/m²) based on length of the day with length of day calculated from latitude and time of year. Calculation of net radiation used saturated vapor pressure (derived from average daily temperature), maximum and minimum daily temperature, total incoming solar radiation calculated in the earlier step, elevation from DEM, and day of year as inputs. Then, reference short crop ET was calculated using net radiation (calculated earlier of elevation), psychometric constant (function of elevation), temperature, actual vapor pressure (from Daymet) as inputs. Wind speed at 2 m above the surface is assumed to be 2m/s, surface albedo is assumed 0.23. The functions to calculate net radiation and reference ET were vectorized to support calculations for 3-dimensional numpy arrays. For the Penman Monteith formulae, a python package called PyETo (Richards, 2015) was used.

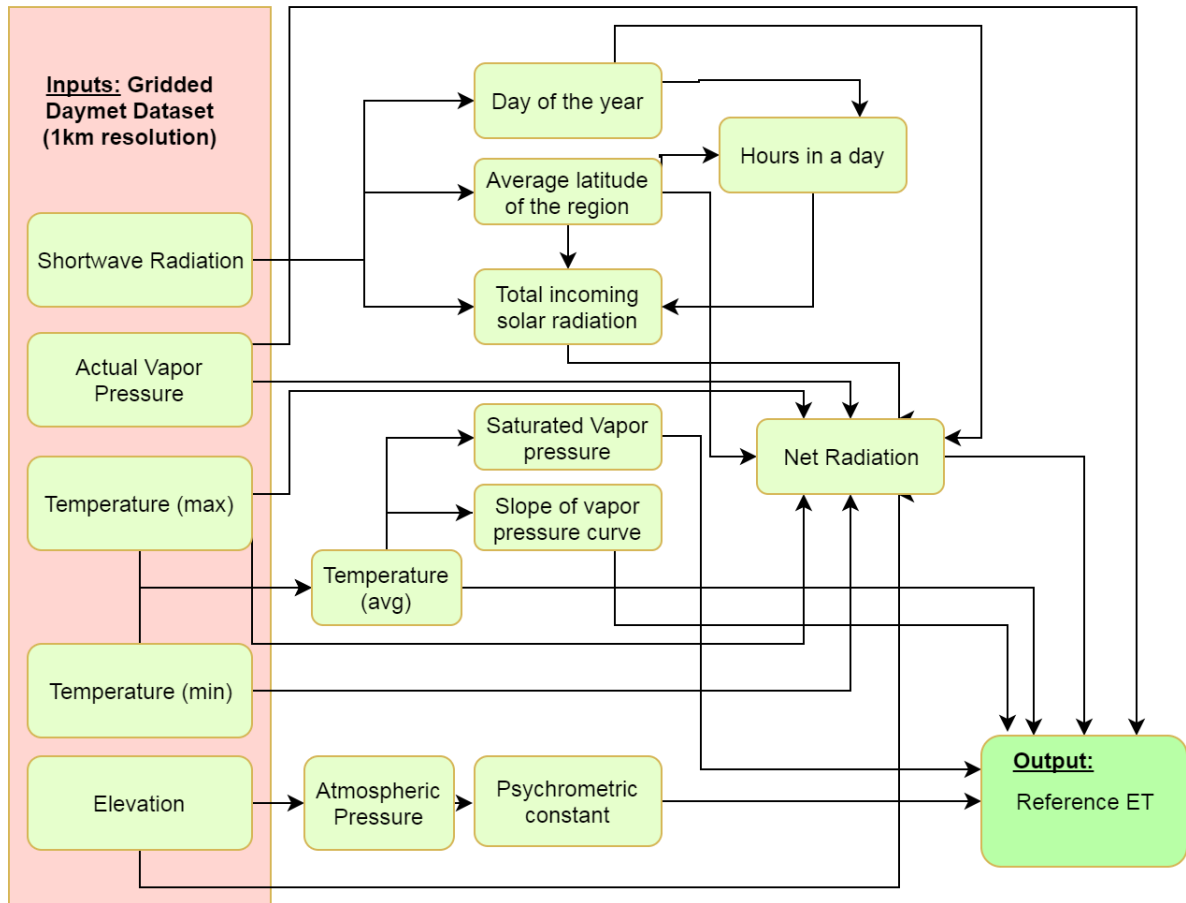


Figure 4. Flowchart showing Penman-Monteith based calculations executed during creation of gridded short crop reference evapotranspiration dataset

3.1.2 Automate TOPKAPI Input-file Preparation

The PyTOPKAPI package used by the app requires an elaborate and particular system of file arrangements to successfully create input files. Python scripts were written to automate the process of creating such files based on the geospatial files (terrain files, soil files, and forcing files). The end result of the automation is a python class and functions that can be used to create and modify a TOPKAPI models as a function of set of geospatial files alone (see HydroDS GitHub page for *pytopkapi_run_instance* class, *runpytopkapi* and *modifypytopkapi* functions in *servicefunctions_pytopkapi.py*).

3.1.3 Retrieve Observed Flow

For proper assessment of model performance, comparison of the simulated flow to the observed flow is desired and error evaluation is required. Hence, a python script (*downloaddailyusgsdischarge* function in *servicefunctions_pytopkapi.py*) was written to fetch observed flow at a USGS gage for a specific time range.

3.1.4 A System to Enhance Users' Access to the Computing Functionalities

The app is designed to provide users access to a variety of computing functionalities. The app is designed in a way that does not restrict users to TOPKAPI modeling. It provides global access to terrain analysis functionalities like watershed delineation, or channel network identification. Additionally, for watersheds in the US, it provides access to SSURGO soil files, climate data (maximum and minimum daily temperature, vapor pressure, short wave radiation), and forcing data (gridded precipitation, and gridded reference evapotranspiration dataset). Appendix F, G, and H give a step by step guide to achieve these functionalities.

3.1.5 GUI Based System for Inputs - Outputs

In order to help users to input the model parameters easily and accurately, a web based system was designed with the use of Graphic User Interface (GUI) elements such as date picker, drop down menus, file uploader, and map elements that accept drawing on them. The overall layout and the style of the app was developed using Tethys Platform (version 1.5). Tethys “gizmos” were used for creating input forms like dropdown menus and date-picker, Highcharts© elements provided by the Tethys Platform were used for

time-series data visualization, and the Maps API from google was used for the map which helped in providing functionalities such as drawing on map.

The app offers its users different functionalities through a set of two web pages: i) home page, and ii) result page. One can also view these pages as a gateway to facilitate connection to specific functions at the HydroDS server, and its interaction with HydroShare.

Accepting User Inputs

A *model_input* web page located at *baseurl/model_input.html* was developed to receive user input. This page contains different web elements to help gather input-parameters required to carry out the user's task. It has two major sections: i) '*app navigation*' section which is located to the left of the screen and contains a variety of forms for taking user inputs, and ii) '*app content*' section containing a google map element which takes most of the screen in the middle (Figure 5). Through *app navigation*, the user can

give inputs to prepare a model or input-files following three steps presented in a set of tabs.

A user can also load a previously saved model through a drop down menu.

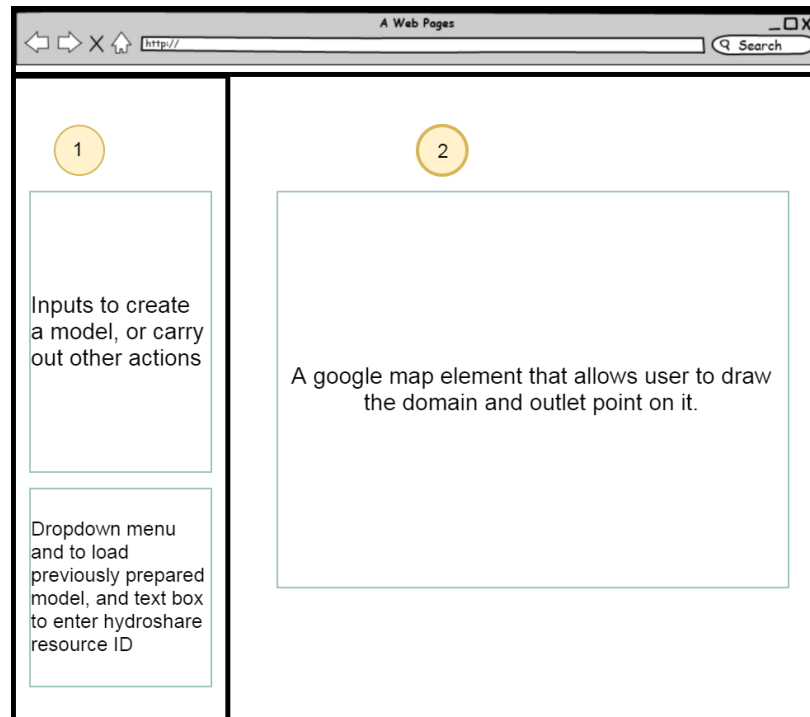


Figure 5. The two major sections in the home page: label (1) is the ‘*app-navigation*’ that contains input forms, and label (2) is ‘*app-content*’ that contains map element.

In the *app navigation* section, there are html-forms for a user to provide parameters necessary to carry out a particular action. The user might define the area in three ways: i) by typing in coordinates for east, west, north and south edges of the domain, ii) by “drawing” (the rectangular area, and the outlet) in the map present in the app, or iii) by providing spatial dataset (e.g. ESRI shapefile, GeoJSON file, or TIFF file) for the domain. There are further inputs such as setting the source for forcing inputs.

Another subsection in the app-navigation section was developed to allow users to load an existing model from a dropdown menu. The section also includes a text box to enter the HydroShare resource ID to load a simulation that may have been prepared by another user and shared in HydroShare.

Displaying the Outputs

The result page located at *baseurl/model_run.html* contains elements to display outputs of the model run (Figure 6). This page contains the same two major sections- *app_content* section displaying the results of simulation prepared including the HydroShare links to the output files created, and *app_navigation* section containing html-forms to fine-tune the model. The results include multiple time-series plots such as simulated and observed hydrographs, rainfall recorded, evapotranspiration leaving the watershed, and the average volume of water in the channel, overland and soil stores in the watershed for each time-step. The graphs are Highcharts© objects provided under the Tethys Platform. The chart element is interactive, and can be zoomed in or out, scrolled back and forth in the timeline, or saved to user's local computer.

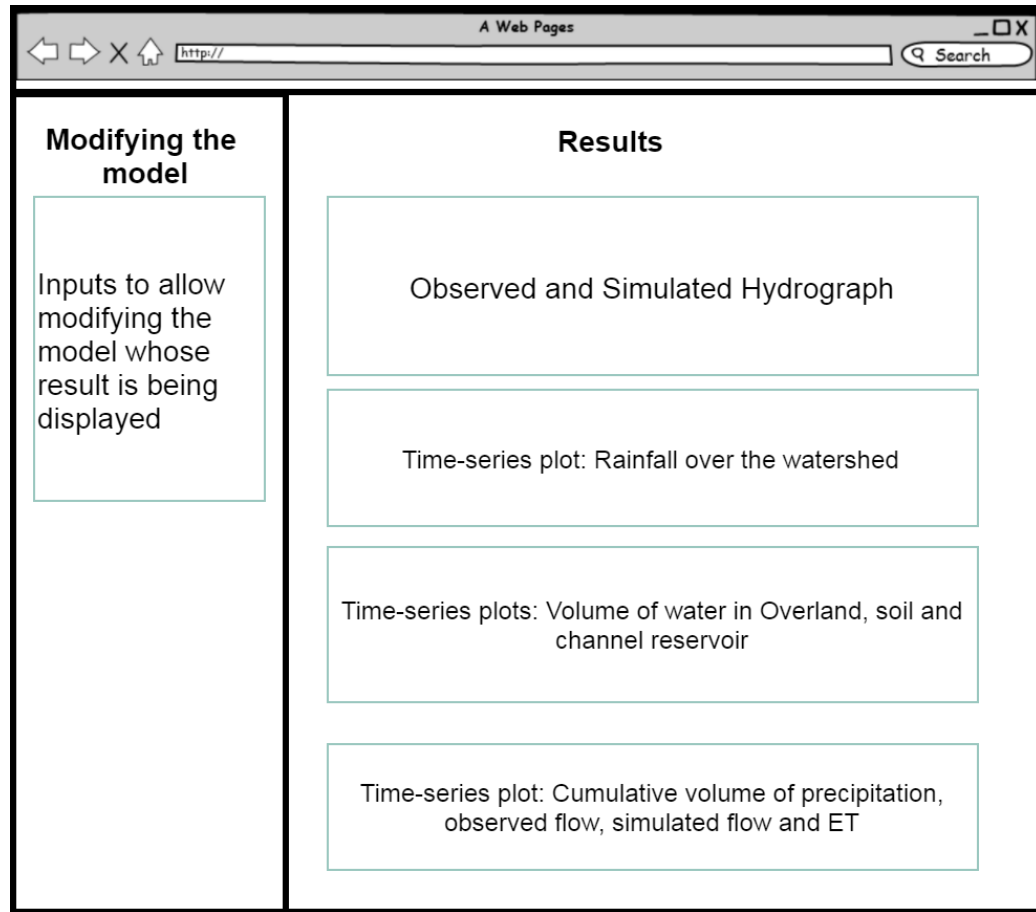


Figure 6. The two major sections in the model-run page: The results section to the left containing Highcharts time-series plots and more, and the section to the right containing html forms to edit the model parameters and re-run the model.

Web Request Architecture

The *model_input* page can create one of the two queries: 1) create a new model or download files, and 2) load previously prepared model. The query is passed onto the *model_run* page. When a user uses the html-forms to modify the result in the *model_run* page, a new query is prepared and is passed onto its own controller, (*model_run()* function). These queries are then used to create a request for HydroDS using the hydrograte python client. Hence, this architecture (Figure 7) allows for the user to modify a run in *model_run*

page, although the run may be prepared for the first time, loaded from a saved models, or just edited from model_run page.

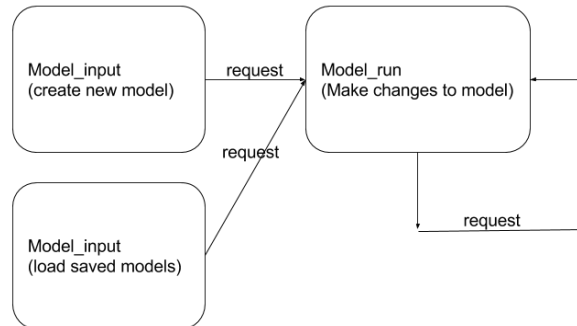


Figure 7. The web request architecture implemented in the app.

3.1.6 The App's Database

One of the functions of the app is to allow user to retrieve a model prepared in the past. For this purpose, the information necessary to retrieve the model is saved in tables in a database. These tables are useful for diagnostic purposes, such as to identify the model's performance in different regions, or with different cell sizes etc. The Tethys PostgreSQL (Momjian, 2001) relational database on the server is used to save and retrieve such information.

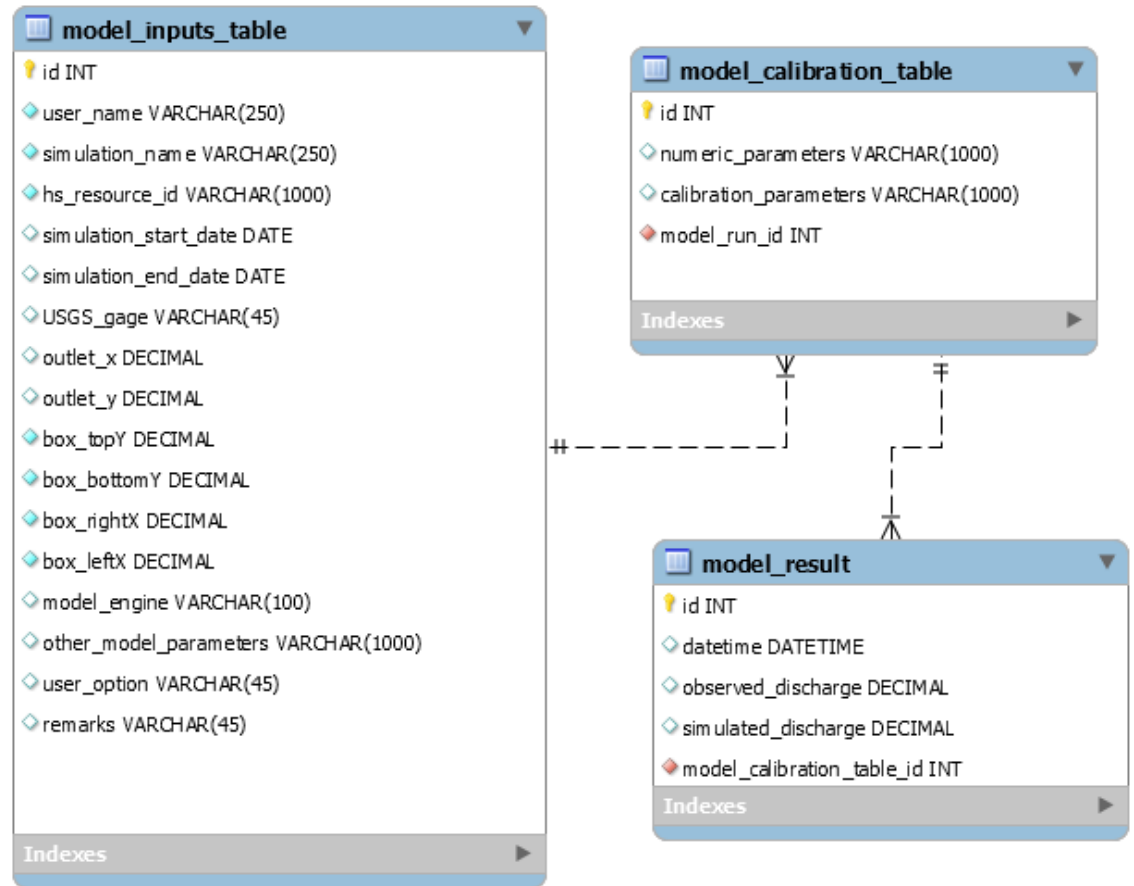


Figure 8. Datamodel used in the app

There are three tables in the database of the app- i) *model_input_table*, ii) *model_calibration_table*, and iii) *model_result* (Figure 8) The information about the model run used to prepare the model such as simulation start and end date, cell size, HydroShare resource ID is saved in *model_input* table; data used to modify the model is saved in *model_calibration_table*; whereas the simulated timeseries data for subsequent model runs is saved in the *model_result* table. The HydroShare resource ID in the *model_input_table* identifies, and can be used to load, for further analysis, a model saved in HydroShare.

The SQLAlchemy (Bayer, 2012) object relational mapper (ORM) which provides an object-oriented wrapper around the SQL interface, allowing for standard database

operations using Python syntax instead of SQL statements, was used to create the database, the tables, update the simulation information, and retrieve the information to load the saved models.

3.2 App Design

This section introduces the design, architecture and the implementation of a prototype web-hosted application (web app) designed to create input files and perform hydrologic simulation on the web.

3.2.1 Overview of the App

The app is designed such that a user does not require any special software or packages and desktop GIS system on the client side, nor does she require any disk space to save the results of the simulation. Access to an internet browser with working internet connection, and a freely created HydroShare user account is enough. The app makes use of files and model sharing functionalities of HydroShare (Tarboton et al., 2014) to take inputs from it, and to save output produced by the app. It uses HydroDS, to carry out the processing of dataset. This system, comprised of the web app, HydroDS and HydroShare can leverage the power of online computing to an end user.

The app accepts inputs from a user, calls a set of functions in HydroDS based on the inputs, and displays results back to user. The app can be used to i) download the geospatial dataset for the domain, ii) prepare input files for TOPKAPI or TOPNET model, and iii) run the TOPKAPI model. If the user wants to run the model (TOPKAPI model), information such as simulation start and end dates, cell size, stream threshold value should be provided. The app will then call functions on the server to process these inputs, and save

the output files, once processed, in the user's HydroShare account. This could take anywhere from a few minutes to few hours.

A brief description of the order of tasks carried out in the background to solve a particular user request to create a TOPKAPI model instance may be broken out into 13 major steps (Figure 9). These are:

Step 1: Input parameters from user are collected from the app UI.

Step 2: The inputs are passed on to model-run page, where its controller function (*model_run()*) calls an appropriate function in the HydroDS. Based on user's request to create a TOPKAPI model, the function *call_runpytopkapi()* from the HydroDS is called.

Step 3: Based on the input parameters, DEM is extracted for the region, and terrain analysis is performed on it.

Step 4: NLCD 2011 map is extracted for the region, and is reclassified based on Kalyanapu (2010) to obtain Manning's n map.

Step 5: SSURGO based soil dataset is derived following a series of steps, including obtaining spatial and tabular information concerning soil properties for the region.

Step 6: Daymet precipitation is extracted for the region based on simulation start and end dates.

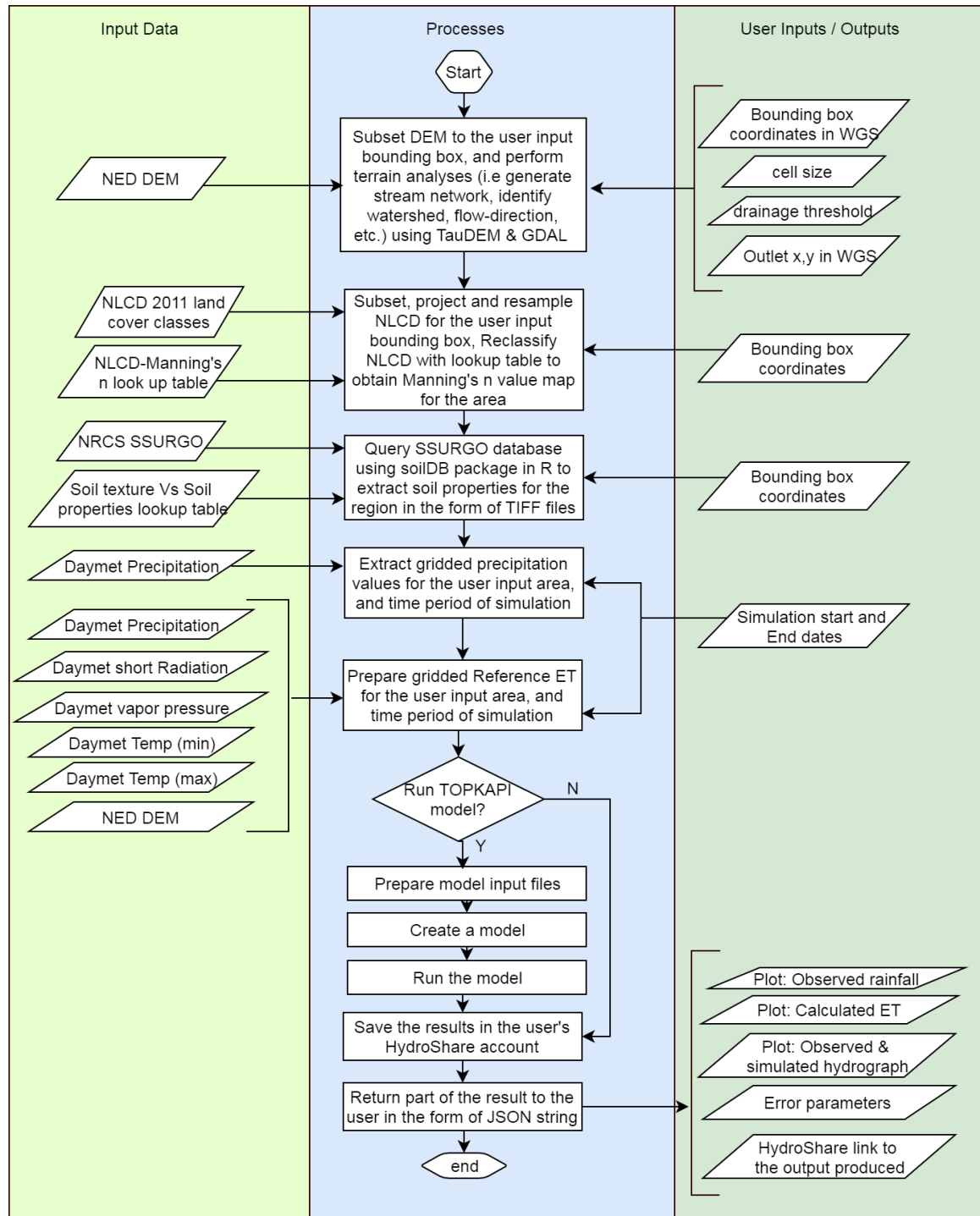


Figure 9. The processes carried out and the inputs used during the execution of a model preparation request by a user

Step 7: Reference ET is calculated for the region by first extracting the Daymet climatic datasets like shortwave radiation, precipitation, vapor pressure, temperature, and

then using Penman Monteith equations based on the guidelines by FAO.

Step 8: TOPKAPI input files are created based on the dataset derived in the steps leading until this.

Step 9: With the input files prepared, the TOPKAPI model is ran.

Step 10: The results of the model execution is collected and saved in user's HydroShare account. Also, the information is summarized in a JSON file format (Appendix E) and sent back to the UI of the app to display it back to the user.

Step 11: When the program receives a response from HydroDS that the model was successfully executed, the information about user's input, model run details, and simulated and observed discharges, are parsed and written in the corresponding tables in the app's.

Step 12 The time-series plots like rainfall, simulated discharges are displayed. Also, the link to HydroShare where the model files and results are saved, is displayed on the screen.

Step 13: In the same screen, users can also make some modifications to model input parameters using the form provided there, and re-run the model. The changes, once submitted, triggers similar actions to when the model is created, but this time instead of creating all the files from scratch, only the changes made to the already prepared input files, and simulation is run. The results are displayed back to the user. Additionally, the app's database is updated to store some of the model run results.

3.2.2 Server Client Architecture

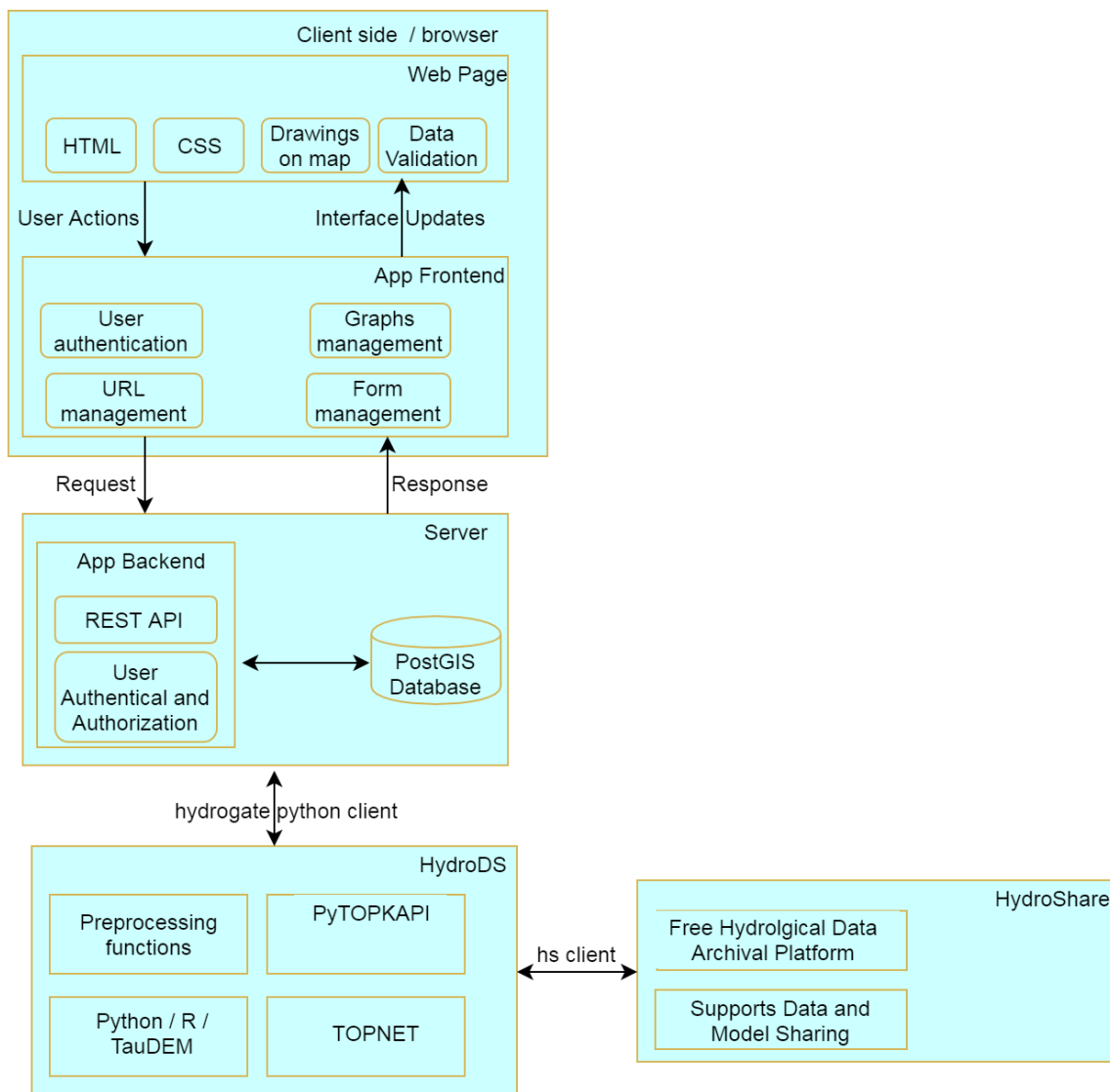


Figure 10. Diagram showing the relationship between the app, HydroDS and HydroShare. Client access the Tethys app through their browser. The backend of the app uses hydrogate python client to call functions in HydroDS server. HydroDS server uses HydroShare python client to save results produced to HydroShare.

Allocating all the processing tasks in the server side can slow down the app. The server and client architecture for the app were designed to distribute the work load to all three components- 1) app's frontend or the client side, 2) app's backend, and 3) the HydroDS web service (Figure 10). Client-side is trusted with some minor processing load

such as warning prompts, data validation, GeoJSON visualization, drawing domain rectangles on the map, searching places on maps. The backend of the app is responsible for tasks such as querying database, processing the parameters collected at the frontend, calling appropriate function from the HydroDS web service, retrieving the response and acting accordingly, processing data in specific formats for plotting time-series plots. The web services are located in a powerful server where many applications and libraries are installed, and hence it makes sense to trust them with the biggest burden of processing datasets. They are responsible for heavier workload such as creating datasets and running the model.

3.3 Experimental Use Cases

This section provides descriptions for the four individual use cases of the app. It talks about the rationale behind the use, important information to consider for each use case, and inputs taken from the user to perform each action. The four subsections respectively describe the four uses cases for 1) the preparation and execution of a TOPKAPI model instance, 2) the preparation of TOPNET input-files, 3) the preparation of hydrologic data, and 4) execution of global terrain analysis.

3.3.1 TOPKAPI Model

A few important considerations have to be made before using the app to model a watershed. First, a user must be confident about the watershed satisfying the model assumptions, particularly the part that says ‘slope of the shallow water table should be reasonably approximated by the ground surface’. Another important consideration is that the precipitation governing the streamflow must be rainfall, not snowfall. Lastly, the

watershed should not be too large (> 400 square kilometers) to avoid large computational time or grid cells that are too big for model assumptions to reasonably apply. Physical quantities such as slope become poorly approximated when grid cells are larger than the scale of topographic variability. Furthermore, there is a computational limit to the size of queries that the SSURGO data service supports, although repeated queries could be used to work around this. Keeping these points in mind, two watersheds were modeled to illustrate the app's ability to prepare a hydrologic model for any watershed in the United States. However, to avoid describing the same procedure twice, only one use case has been discussed in the thesis, while the procedures and the results for the other has been moved to Appendix E, where it can serve as a tutorial for users on how to use the app to prepare a TOPKAPI model instance.

The first watershed selected, the one we will be discussing, is located north-west of Santa Barbara in California (Figure 11). The watershed drains into Santa Cruz Bay of the artificial Lake Cachuma. The app was used to create a TOPKAPI model that might be used for answering a multitude of hydrologic questions, such as understanding the response of the watershed to some forecasted forcing, estimating actual evapotranspiration from the surface, or estimating impacts of landuse changes in streamflow etc. Here, we demonstrate the preparation of a TOPKAPI model, and use hydrological judgements to modify the model inputs using the functionality provided by the app.

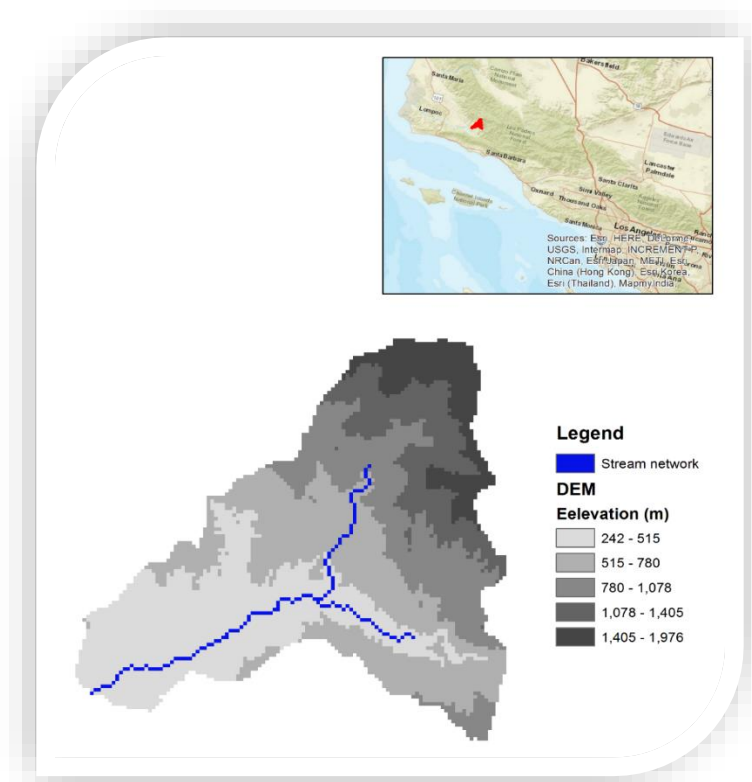


Figure 11. Study site location and DEM of Santa Cruz watershed

The input parameters required to prepare the model were obtained with the help of the app map interface to draw outlet points and model domain, and the texts box and different ‘gizmos’ to enter other inputs. Although the app supports cell size as small as 30 meters, a cell size of 100 meters was chosen for the region to limit the computational burdening for the 72 square km watershed. A stream threshold of 5 square km was used to define a channel cell. The discharge data from nearby USGS gage (ID 11124500) was used to compare the simulated hydrograph with the observed one. Table 6 lists all the input information provided to create the model.

Table 6. The input parameters used for creating the TOPKAPI model

Inputs From User	Values	Units
Simulation Name	SantaCruz_demo	String
USGS gage nearby	11124500	String
Simulation start date	10/01/2010	mm/dd/yyyy

Simulation end date	11/01/2010	mm/dd/yyyy
Stream threshold	5	km ²
Cell size	100	Meters
Outlet location latitude	34.59637	Degrees (WGS84)
Outlet location longitude	-119.90873	“
Bounding box minimum latitude	34.714	“
Bounding box minimum longitude	34.586	“
Bounding box maximum latitude	-119.90873	“
Bounding box maximum longitude	-119.925	“

The process to create model includes two major steps: 1) creating input files for terrain, soil properties, and forcing in a gridded format, and 2) using those files to run the model.

3.3.2 TOPNET Model

Logan watershed, located in the north-east part of Utah, was selected to illustrate the app's ability in preparing model-input files for TOPNET, using the input parameters in Table 7. As noted above, the app does not support running the TOPNET model. Instead users would need to set up and run TOPNET following methods of prior work by others Sazib(Sazib, 2016), and Tarboton (2007a; 2007b). At its present stage of development the app can only serve the purpose of helping expedite the process of creating TOPNET model inputs.

Table 7. The input parameters used for creating input files for TOPNET model

Inputs From User	Values	Units
Simulation Name	Logan_TOPNET_2015	String
USGS gage nearby	10109000	String
Simulation start date	01/01/2015	mm/dd/yyyy
Simulation end date	12/30/2015	mm/dd/yyyy
Stream threshold value to identify stream network to align outlet to it	100	Cell count

Minimum threshold value used in Drop analysis selection of stream delineation threshold	500	“
Maximum threshold value used in Drop analysis selection of stream delineation threshold	5000	“
Number of stream threshold values used in Drop analysis selection of stream delineation threshold	12	
Cell size	30	meters
Outlet location latitude	41.7436	Degrees (WGS84)
Outlet location longitude	-111.7836	“
Bounding box minimum latitude	-111.83	“
Bounding box minimum longitude	42.12	“
Bounding box maximum latitude	-111.44	“
Bounding box maximum longitude	41.68	“

3.3.3 Download Hydrologic Dataset

Mill Creek watershed, located in the north-east part of Illinois, was selected to illustrate the app's ability to prepare hydrologic files for any part of US. The dataset that include terrain files, soil properties files, and climate / forcing files may be used to create some other hydrologic models, or for other purposes. Table 8 lists all the input information provided.

Table 8. The input parameters used in downloading hydrologic dataset

Inputs From User	Values	Units
Simulation Name	MillCreek_IL_2010	String
USGS gage nearby	05527950	String
Simulation start date	10/01/2010	mm/dd/yyyy
Simulation end date	10/01/2011	mm/dd/yyyy
Stream threshold	2	km ²
Cell size	100	Meters
Outlet location latitude	42.4153	Degrees (WGS84)
Outlet location longitude	-87.9692	“
Bounding box minimum latitude	-88.089	“
Bounding box minimum longitude	42.554	“
Bounding box maximum latitude	-87.962	“
Bounding box maximum longitude	42.30	“
Want terrain dataset	Yes	Boolean (check box)

Want soil dataset	Yes	“
Want climate/forcing dataset	Yes	“

3.3.4 Perform Global Terrain Analysis

An outlet point in the Bagmati River that runs through the Kathmandu Valley of Nepal was selected to illustrate the app’s ability to perform terrain analysis anywhere on earth (subject to size limitations). The output dataset from the analysis include pit-filled DEM, flow direction map, slope map, watershed map. Table 9 lists all the input information provided to create the model.

Table 9. A list of information given for performing terrain analysis

Inputs From User	Values	Units
Simulation Name	BagmatiRiver-Ktm	String
Stream threshold	5	km ²
Cell size	50	Meters
Outlet location latitude	27.6121	Degrees (WGS84)
Outlet location longitude	85.2905	“
Bounding box minimum latitude	85.1166	“
Bounding box minimum longitude	27.89	“
Bounding box maximum latitude	85.55	“
Bounding box maximum longitude	27.5	“
EPSG code for projection	32644	Integer, UTM Zone 44

4. RESULTS

In this section, we present the results of the work, which are the web-app, and the simulations conducted using it. The simulations illustrate the utility of the web app for modeling and for retrieving hydrologic data. This section is organized into two segments: In the first segment, the screenshots of the app's user interface and design are presented. The second segment presents the results of experimental use cases using the web app.

4.1 Software Implementation

The software was implemented based on the software-design mentioned above. This resulted in two dynamic web pages to perform the functionalities offered by the app- a model input page to collect inputs from users, and a model run page to display results.

The home page or model-input page (Figure 12) shows its three sections. First, the section located to the left of the screen contains HTML forms and Tethys 'gizmos' to collect parameters necessary to create models (Number 1 in Figure 12). Second section located right below the first (Number 2 in Figure 12) has a dropdown menu to load previously prepared simulations. The section also includes a text box to enter the HydroShare resource ID to load the simulation prepared by other users but made public. Third section occupies the most screen space, and is used to allow users to draw the domain and outlet point on the map.

Inputs for the model

Prepare the hydrologic model by entering the inputs (in the three steps) below.

1. General Inputs

Simulation name

USGS gage nearby

Start Date **End Date**

2. The domain

3. Additional Inputs

[Only interested in input files? Click here!](#)

Simulation already prepared? Load the previously saved simulation from the dropdown menu, or enter the HydroShare resource ID containing simulations files.
Saved Models

Enter HydroShare resource id instead

Specify the outlet, and the bounding box that completely contains the watershed

This is testing area for prompt..

Map **Satellite**

2. The domain

☐ Specify long lat, or draw on map

OUTLET
Longitude **Latitude**

BOUNDING BOX
North Y

West X **East X**

South Y

☐ Upload shp files or TIF

☒ Upload geojson files for watershed

[Click here to Upload](#)

Figure 12. Model input page of the app showing the three sections: (1) Input-forms to prepare a model, (2) gizmos to load existing model, and (3) map element that supports drawing domain and outlet point on it

The model-run page (Figure 13) contains two major sections. The first section is the result section (Number 1 in Figure 13), occupying the most space on screen that displays the results after running the simulation. This section contains results such as observed and simulated discharge plots, rainfall plot, and other time series plots that were results of the model run. These plots are Highchart elements that a user can zoom into, or download to their local computer in multiple formats. The second section (Number 2 in Figure 13), is located to the left of the screen and contains HTML forms. These forms can be used to input parameters to modify the simulation being loaded. Once the modifications

are submitted, the result from the model run created using the modified model parameters are displayed back to the user in the same screen. Users can make multiple modifications, and the results will be displayed in the same model-run page. Also, the results files will be updated in the HydroShare Resource.

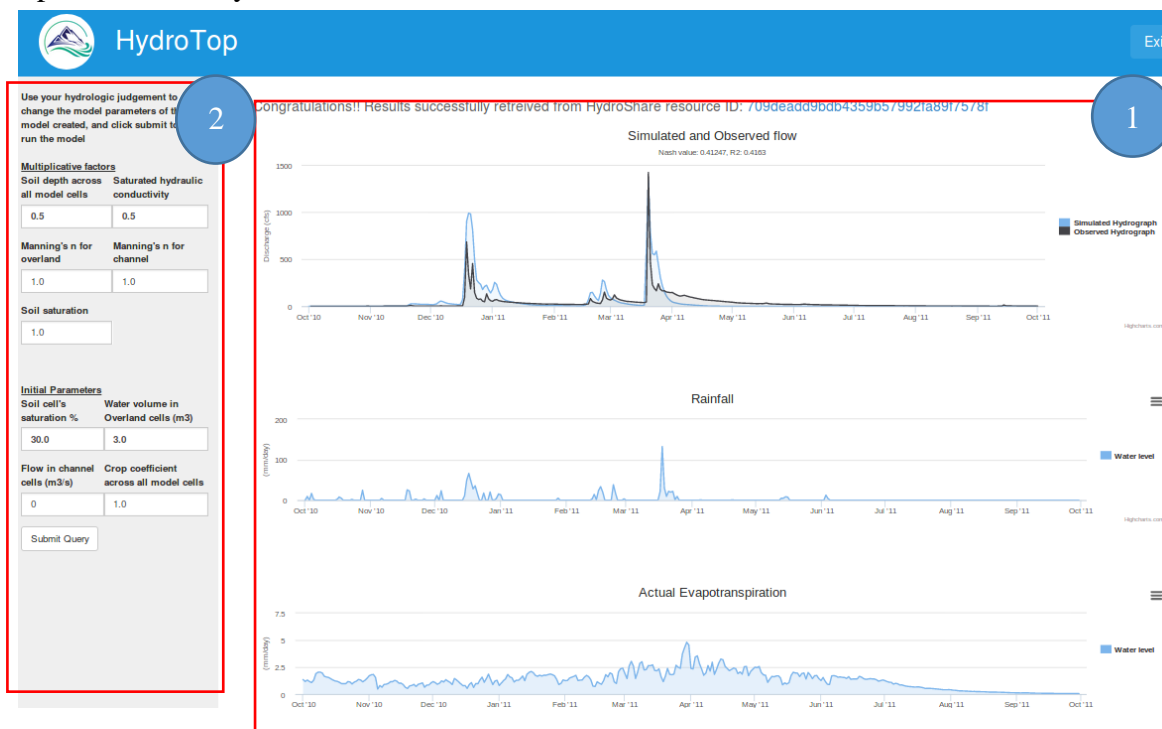


Figure 13. Model-run page of the app showing the two sections: (1) display results of the model run, and (2) Input-forms to modify the loaded model

4.2 Use Case Results

4.2.1 TOPKAPI Model Results

The app prepared a TOPKAPI model for the Santa Cruz watershed, calculated the runoff at the user specified outlet point, and created other time-series plots. Using the app, a comparison between simulated hydrograph and observed hydrograph was also possible (Figure 14). For comparison purpose, the observed flow information for the USGS gage 11124500 was downloaded from the USGS website on the fly. Time-series plots of actual (ET), rainfall, volume of water in soil, overland and channel reservoir etc. were also created

(Number (a) through (e) respectively in Figure 15). The app also created a cumulative graphs of rainfall entering the watershed, channel flow leaving the watershed and ET for the time period of the simulation (Number (b) in Figure 15). The first model run with parameters determined directly from the available data resulted in Nash–Sutcliffe Efficiency (NSE) coefficient for the observed and simulated discharges of 0.36, while the R^2 value was obtained to be 0.42. The app reports these. By hydrologic modeling standards a NSE value of 0.36 is not good and a modeler would typically work to improve this by adjusting parameters through manual or automated calibration, or through revision of inputs where they are deemed to contribute to the differences. Although the app supports basic modification, which will be demonstrated shortly, full automated calibration was beyond the scope of what I was able to implement in this app, but one advantage of the app being interoperable with HydroShare is that automated calibration developed by someone else as part of HydroShare could interoperate with and exploit results from my app. The plot for cumulative volume of water in the basin is also a part of the result, as shown in the figure 16.

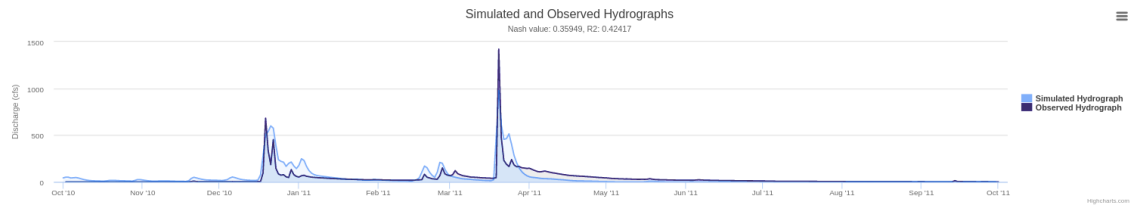


Figure 14. Result provided after the model run: simulated and observed hydrograph

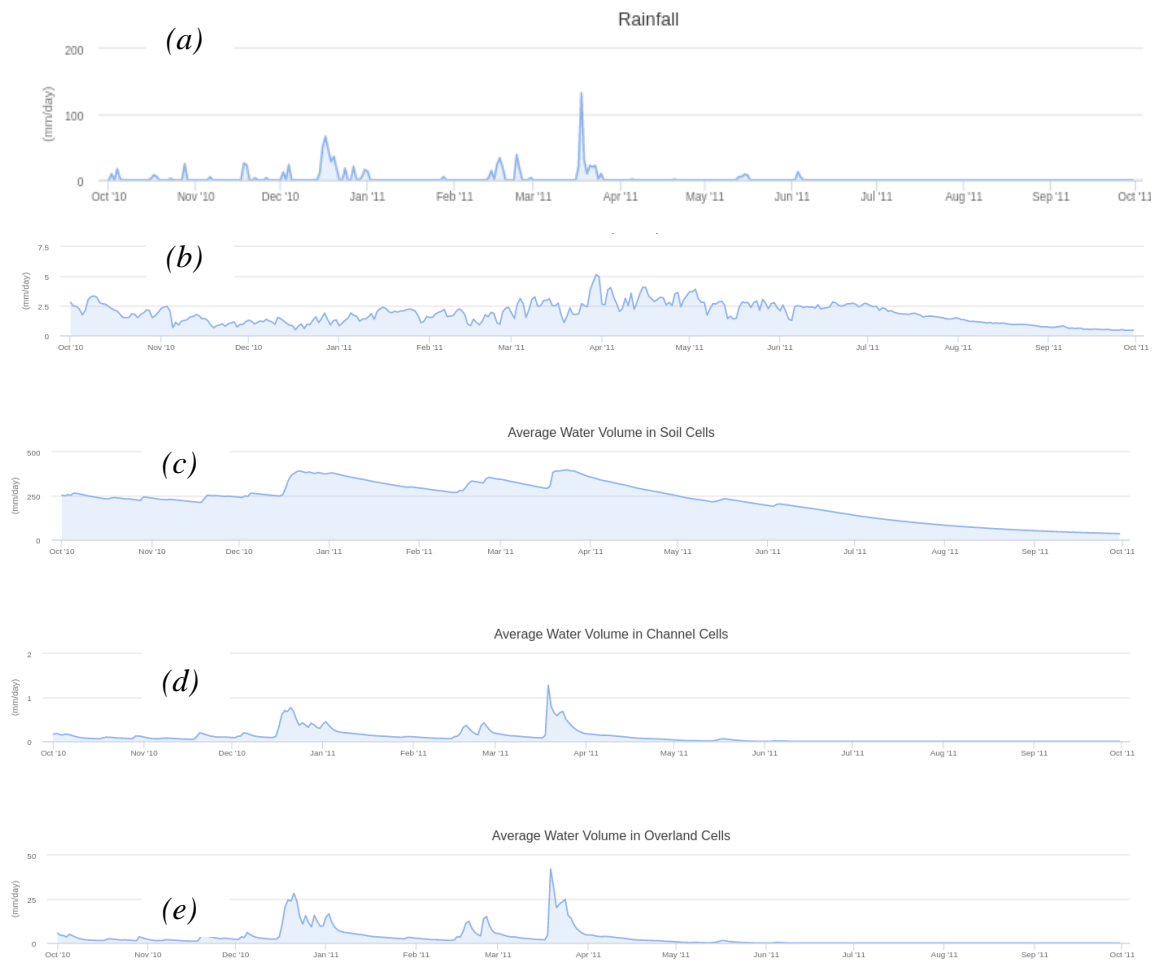


Figure 15. Result provided after the model run: Time-series plots of (a) observed rainfall (input), (b) evapotranspiration, (c) water volume in soil cells in mm, (d) water volume in channel cells, (e) water volume in overland cells in the watershed. (Note that this is a screen shot from the app to illustrate results. Axis labels visible on the screen do not display at the scale of printing in this figure)

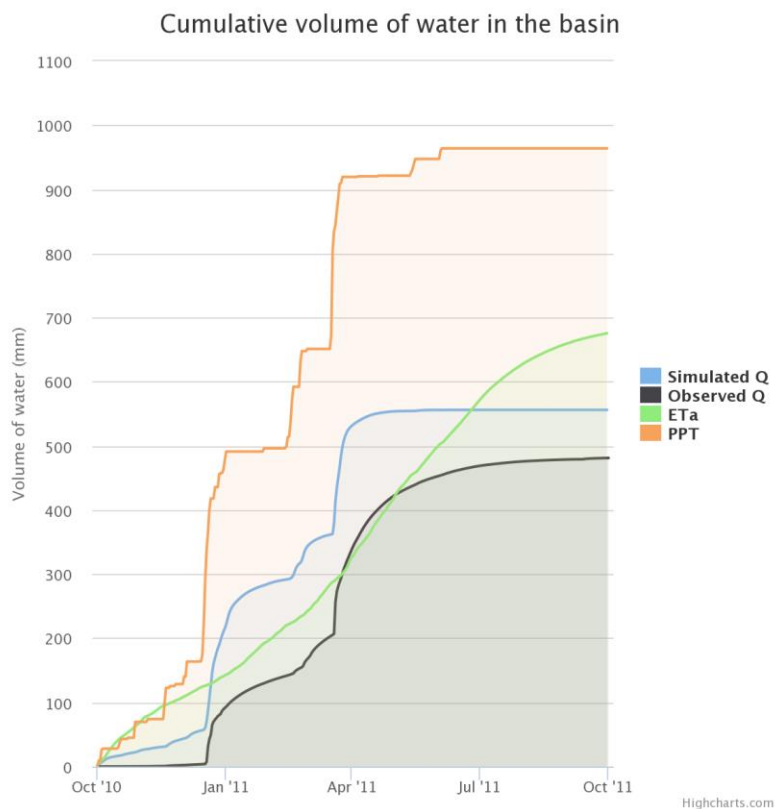


Figure 16. Result provided after the model run: Cumulative volume of water entering and leaving the watershed

Calibration Parameters	
fac_L	fac_Ks
1.0	1.0
fac_n_o	fac_n_c
1.0	1.0
fac_th_s	
1.0	
Numeric Parameters	
pvs_t0	vo_t0
30.0	3.0
qc_t0	kc
	1.0
Submit	

Figure 17. Initial parameters value used while creating the model

The app supports making modifications to the model. Users can use hydrologic judgement to modify the model by changing initial conditions like soil saturation, volume of water in overland or channel soil etc., or altering multiplying factors for crucial catchment properties such as saturated hydraulic conductivity, soil depth, Manning's coefficient for overland or channel flow etc. In our case, we see that the model in the beginning period overestimates the flow, although there seems to be very little rainfall. This suggests that the initial soil saturation that was in the app setting was too high for this case. This was the reason for the early occurrence of discrepancy in observed and simulated cumulative water volume, which adds up until the end of the graph as shown by Figure 14. Based on these considerations I changed the initial saturation from 30% to 15%. Another consideration is that the flow, in general, seems to appear slightly ahead of the observed flow. This suggests the routing is too fast. Hence, I also decrease the multiplicative factor for routing, from 1 to 0.5. Third, note that hydrograph peaks, especially the last one, are underestimated by the model. This suggests that the soil depth of the unsaturated zone is larger than it should have been. Hence, I adjusted the multiplicative factor for soil depth down to 0.7 from 1. With these changes (summarized in Figure 17), the model was re-run. This resulted in a modified model with changed parameters and the final results are similar time-series graphs as running the model for the first time. The Nash–Sutcliffe coefficient and R^2 values for the observed and simulated discharges for the modified model (Number (a) in Figure 18) increased to 0.52 and 0.52 respectively, indicating an improvement that the analyst/user can achieve through interaction with the app. The modifications also improved the volume balance of the system. There was also a smaller error in the volume of water leaving the watershed compared to observed values (Number (b) in Figure 18).

with the error in volume calculations dropping from +18% to +2% under the modified model result.

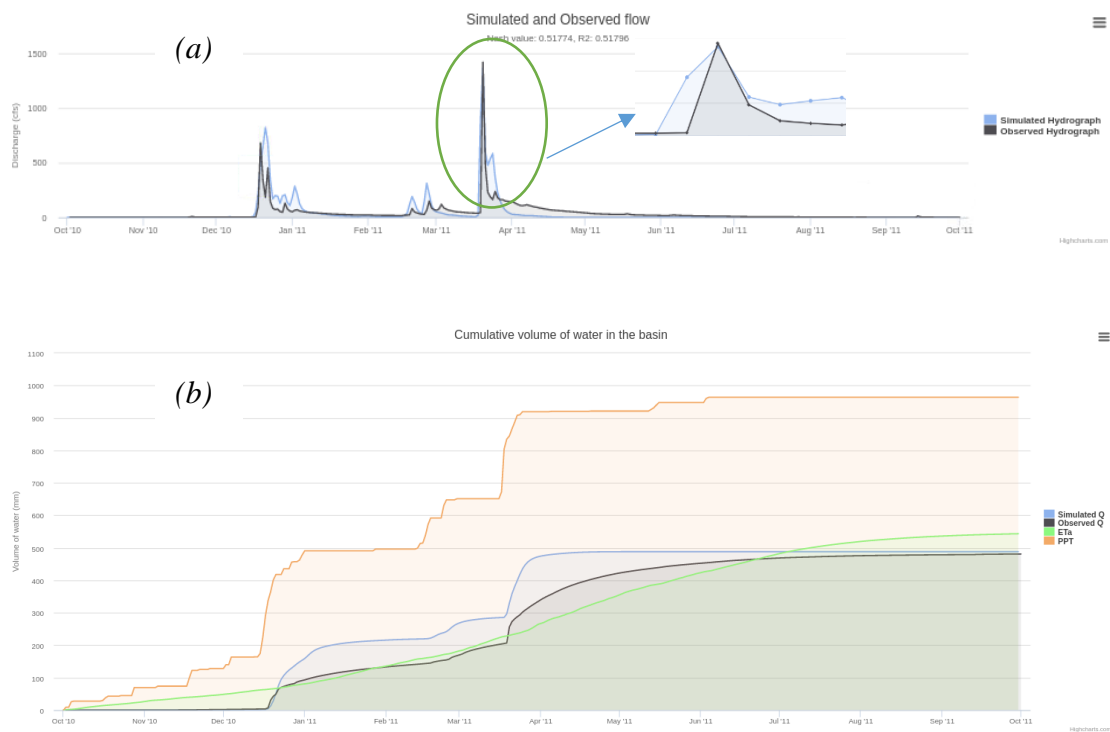


Figure 18. Results obtained after rerunning the model with a changed parameters value. In part (a) of the figure, we can see the improvements in the estimation of peak discharges, especially one observed in April, compared to the original simulation. Also in part (b) we can see a simulated water volume leaving the system is much closer to observed than in the first

Multiplicative factors	
Soil depth across all model cells	Saturated hydraulic conductivity
0.7	1.0
Manning's n for overland	Manning's n for channel
0.5	0.5
Soil saturation	
1.0	
Initial Parameters	
Soil cell's saturation %	Water volume in Overland cells (m3)
15.0	3.0
Flow in channel cells (m3/s)	Crop coefficient across all model cells
0.1	1.0
Submit	

Figure 19. Changed initial parameters value used to modify the model

The entire model run information-including the inputs used to create the model, and the outputs are summarized in a text file in the form of JSON dictionary. This file (see Appendix E to see how it is structured) is also a part of result that is saved in HydroShare. This file not only serves as record of input and output files, but also accommodates multiple run information including the time series data. Both the model has been made public, and can be found in HydroShare under URL: <https://www.hydroshare.org/resource/9612dfd6a50c4a9194d9ed7e8956f555>

4.2.2 Prepare Model-input Files for TOPNET

The app prepared input files for a TOPNET model instance for the Logan watershed. The DEM and NLCD for the region was extracted, projected and resampled. Peuker Douglass watershed delineation was performed for the given outlet based on the thresholds given by the user, Daymet climate data was downloaded on the fly for the domain. Nodelinks and reachlinks information is generated for the watershed and the wetness distribution is created. After that, soil dataset is prepared, followed by creation of basin parameters and rain-weight file using PRISM dataset. Finally, observed streamflow time series is downloaded. Note that the result here is the capability for the app to execute a model input preparation workflow and produce model input files equivalent to what previously required tedious and platform dependent desktop execution. I did not develop the workflow, but did enable it through the app. For detailed information on the workflow and the descriptions of data it produces, readers are referred to Sazib (2016) and Tarboton (2007a; 2007b)

The results obtained were saved in HydroShare. The links to the HydroShare resource, and to the zipped folder containing the prepared files are displayed to the user in the app (Figure 20). The HydroShare resource containing the result of this experimental case has been made public, and is available at:

<https://www.hydroshare.org/resource/955d72a9b00141548aca582c493ee140>

Congratulations! Files were created successfully!

The created geospatial and forcing files can be accessed temporarily by using this link: [Download zip here](#)

The files is also saved permanently in HydroShare, with the ID [955d72a9b00141548aca582c493ee140](#)

Figure 20. Screenshot of the links to the results displayed by the app after TOPNET input-file preparation

4.2.3 Download Hydrologic Dataset

For the Mill Creek watershed three set of hydrologic files containing terrain, soil and forcing data were prepared based on the user's request. The files created have been listed (Table 10), and visualized (Figure 21). Once the analysis was complete, all the files were zipped and the link to the zipped file was displayed to user in the output screen. Also, the files created during the process were saved in HydroShare. The HydroShare resource containing the result of this experimental case has been made public, and is available at:

<https://www.hydroshare.org/resource/a3552bea9df243be822a61349a200ece/>

Table 10. A list of hydrologic dataset created

Files created	Type	Source	Process
DEM	Terrain dataset	NED	TauDEM assisted analysis
Ground Slope (tan B)			
Stream network			
Flow direction (D-8)			
Watershed			
Depth of surface soil layer (m)	Soil	NRCS	SSURGO dataset processing
Saturated hydraulic conductivity			
Residual soil moisture content			
Saturated soil moisture content			
Bubbling pressure			

Pore size distribution			
Drainable porosity			
Plant available porosity			
NLCD	Land cover	USGS	NLCD to Manning's n look up table
Manning's surface roughness coefficient			
Rainfall	Forcing	Daymet	FAO Penman - Monteith guidelines
Reference Evapotranspiration			
Maximum, minimum daily temperature	Climate		NetCDF subsetting
Vapor pressure			
Short wave radiation			

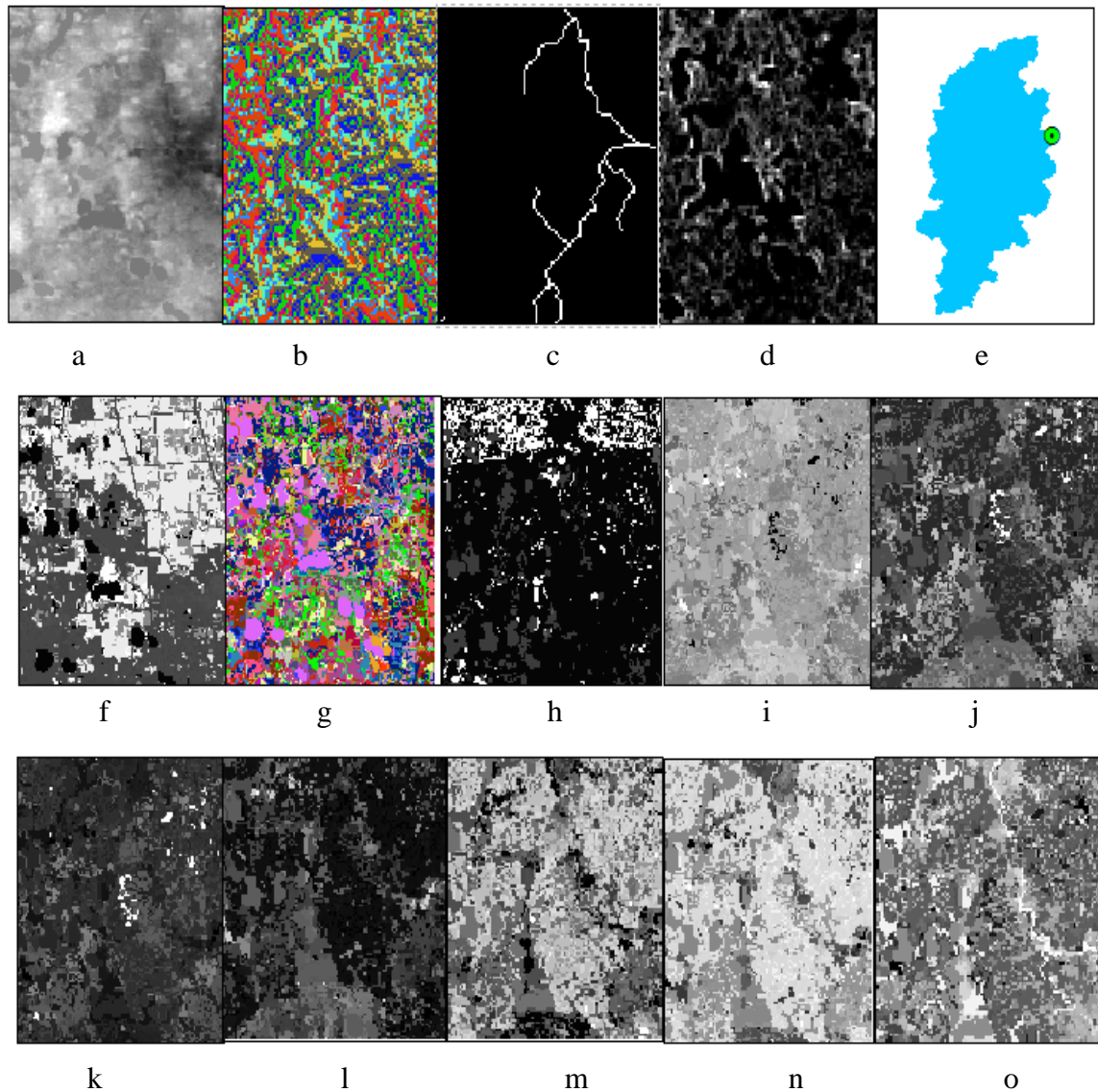


Figure 21. Some of the hydrologic dataset created for the Mill Creek watershed- a. DEM, b. flow direction, c. stream, d. Slope, e. watershed, f. NLCD, g. map unit keys, h. soil depth, i. drainable porosity, j. wilting point, k. wetting front suction, l. pore-size distribution, m. porosity, n. residual moisture, o. suction

4.2.4 Perform Global Terrain Analysis

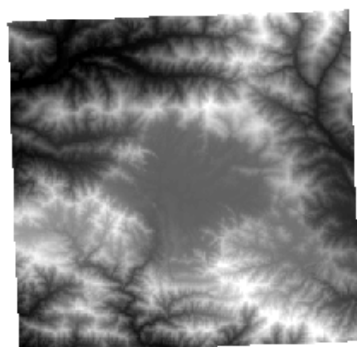
The terrain analysis was performed for the Bagmati River watershed as explained in the dataset preparation subsection earlier. Table 12 and Figure 22 shows the output terrain files created by the analysis. All the files created were zipped and the link to the

zipped file was displayed to user in the output screen. Also, the files created during the process were saved in HydroShare. The HydroShare resource containing the result of this experimental case has been made public, and is available at:

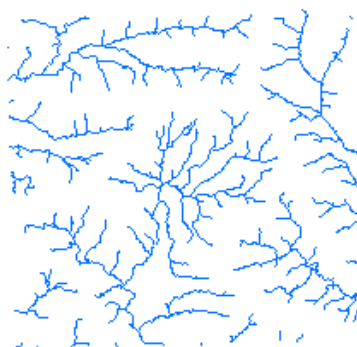
<https://www.hydroshare.org/resource/7c54189ea9ed4ac7b6eb73b1d6ec985d/>

Table 11. A list of terrain dataset created

Files created	Type	Source	Process
DEM	Raster	SRTM	TauDEM assisted analysis
Stream network			
Ground Slope (tan B)			
Flow direction (D-8)			
Flow accumulation			
Watershed			
Watershed	ESRI Shapefile	User input	
Corrected outlet	ESRI Shapefile		



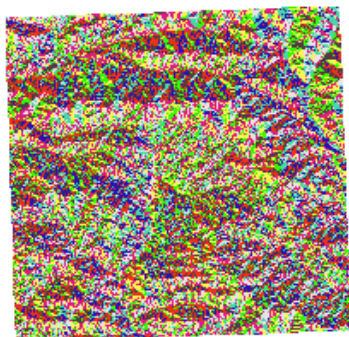
a. DEM



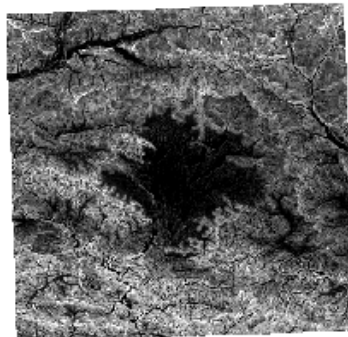
b. Stream Network



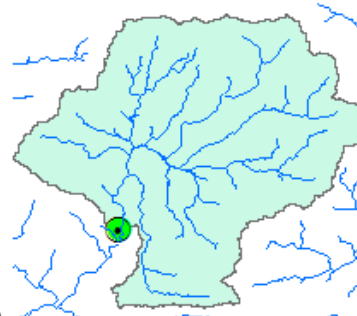
c. Flow accumulation



d. Flow direction



e. Slope



f. Watershed

Fig 22. Some of the terrain dataset created for the Bagmati River at Kathmandu

5. DISCUSSION

The thesis addresses the data related challenges involved in hydrologic modeling with the development of a prototype web application. By implementing a web based solution, the intensive data preprocessing and modeling functionalities are moved to powerful servers away from the end users and the model and data sharing capabilities of HydroShare are exploited. This approach not only makes the app machine independent, it also lowers the data and computing related barriers in modeling by giving end users access to modeling and data processing functionalities. By providing modeling and data access as web services, this work contributes to the vision of Model as a Service mentioned in the background section. We begin with a discussion of the significances of the work, then explore the relation of the approach to existing work, and follow with a more general discussion of strength, limitations and future improvements of the work.

For practical purpose, the app can be used to rapidly- i) create a TOKAPI model for any watershed in the US, ii) create TOPNET input-files for western US (a limitation due to hosting of some required data), iii) get terrain, soil or forcing datasets for anywhere in US, or iv) perform terrain analysis anywhere in the globe. These functionalities are useful to a hydrologist to quickly prepare a hydrologic model, which helps him better allocate his time and resources for the more important analysis part rather than on preparing the model. But I believe the importance of the system goes beyond that. The simplicity of the process can attract newer user groups. For example, an undergraduate civil engineering student with transportation major could use the app to simulate flow on a stretch of road he is designing for a project, and see the effect of change in initial conditions on the flow. The app can be a valuable resource for students and inexperienced modelers not only to

better understand rainfall runoff processes, but also to help spread the science of hydrology with different scientific groups.

Compared to the existing approach of modeling in desktop computers, the web based approach provides a number of benefits like improved usability, accessibility, and maintainability. A typical desktop modeling approach involves manually getting the required dataset from internet or other sources, using expensive and machine dependent GIS Desktop software to preprocess the dataset, using difficult programming scripts or tedious manual labor to prepare input files, and running the model. Compare that to the web based solution which is better in terms of i) usability; it is simpler to use, offers collaboration functionalities with other people, offers to save dataset on the cloud, etc. ii) accessibility because it can be accessed by multiple users at once from a variety of devices, from any place in the world that has a working internet connection, and iii) maintainability because it does not require a user to upgrade software periodically, etc. However, this approach, too, is not free from limitations. The most crucial limitation is it requires a working internet connection. Other limitations stem from the app being connected to server to function. For example, the servers have a limit on number of users they can accept at a time, which when exceeded, can shut down the server completely. If the server is not running properly for any other reason, it can render the app non-functional thereby affecting all the users.

There have been similar web based works in the past for both data and model access, however, many of them come with their own limitations. For example, SWATShare (Rajib et al., 2016) allows users to share their SWAT model online to other collaborators, make multiple runs altering input parameters, and also produces excellent varieties of plots.

However, the users will have to prepare their own models first and upload them to the app. When contrasted to established system like HydroTerre (Leonard and Duffy, 2013), the types of data the app provides is somewhat restricted, but the upsides are that the users are not restricted to level-12 HUC units for data, and the app also supports model-runs using the data. Works by Choi (2005), and Granell (2010) implemented similar systems but by using empirical models requiring fewer parameters, and lumped models respectively. However, none of the previously mentioned applications provide web based modeling involving physically based fully distributed models that require extensive parametrization.

Apart from the strengths as a result of being a web app, there are additional strengths of my app in particular. The biggest strength, which has been mentioned before, is providing users with access to hydrologic data, and to modeling functionalities. The app records the input parameters used, the initial values given/estimated, the important time series results etc. in a JSON file. This serves as both a record, and can be used to recreate the model. The input system uses GUIs and supports uploading of multiple file formats to simplify process of accepting inputs, the output uses Highchart plots for better visualization. Results are saved on the cloud, previously prepared models are saved for future use, and model result can be modified. SSURGO based soil files and Daymet forcing files are available to download for the entire US. Terrain analysis is available from all over the world. The biggest strength of the app might well be that it is free and open source which invites general public to participate in improvements and creative usage.

On the other hand, the app also has several limitations, most of which are inherited from limitations of the data itself. There are different agencies that employ different methods to record same type of dataset. The app only makes use of a particular set of data

sources. For example, for one particular location there might be a governmental agency recording precipitation information using radar, and a university research team recording same precipitation using gages. The quality and suitability of these dataset vary, hence in general it is sensible to use one over the other. Since the dataset source used in the app is restricted, it may not be the ideal dataset for certain cases. It all comes down to the accuracy, the level of quality assurance/ quality check that has been applied to the dataset, the error and uncertainty of the results associated with the input dataset. The app neither makes its analysis considering these factors, nor provides options for users wishing to use his/her own dataset. Dataset such as DEM for non-western-US and soil dataset are prepared by querying data services on the fly, hence the entire functionality of the app is tied to the successful operation of such queries. Also, there is data limit imposed on those queries, which becomes data limit for the app. For example soil dataset query is restricted to 100,000 acres or 100,000 table records. Also, if those agencies change their policies, or the rules in their APIs, the entire app could be nonfunctional. It is, however, common that the providers retain the old rules for a period of time to allow developers to make changes accordingly. As long as a developer is looking after the app, the limitation could be avoided. For forcing dataset, only daily records is supported, only for the time period from 2005 to 2015. Another weakness of the app is that it does not contain snow modeling, hence it only supports watersheds where major precipitation source is rainfall. Additionally, being a distributed model, there is also computational limitations. However, the computational limitations have not been quantified because the dataset limitation restrict the app before it does (i.e. the combination of limitations of cell size $> 30\text{m}$ resolution, watershed area $< 100,000$ acres and forcing dataset range < 10 years restrict the app before the computational

restrictions come along). Many of these limitations also exist in the conventional desktop based modeling practices. Despite these limitations, the app demonstrates a strong cases for the use web based modeling solution to tackle challenges in hydrologic modeling.

There are factors that affect the performance of the app, although strictly speaking they are not truly the limitation of the app or of model. The most important point to consider is that the watershed should comply with the model assumptions mentioned in the background section. Then, the next most important thing would be proper initial conditions of the watershed. Although the app selects certain set of initial parameters, they may not be reliable. Other factors include the selection of cell size, which should not be too large that the ground features get lost, or the selection of stream threshold value should not be unreasonable that they stream network formed is unreal. In general, model is a garbage in garbage out system- bad inputs will result in bad outputs, hence proper care must be taken.

While this thesis has demonstrated the potential for web based system for providing users with access to data and modeling functionalities, many opportunities for extending the scope of this thesis remain. We will begin with a disclaimer that not every hydrologic model can be implemented as a web app. It depends on factors such as complexity of the model or the data it requires. The app developed was based on free and open source philosophy, hence, future works are welcome to use and make additions to the app. The app can serve to be a structure for GUI, results visualizations, data preprocessing, and data storage for future apps. Future works could focus on improvements in data sources-like implementing high resolution DEM, high resolution (both spatial and temporal) forcing dataset etc. For models such as TOPNET, a complete modeling functionality would be a good addition to the input-file preparation that has already been done. More models could

be added to the app, allowing users not only to choose one over the other, but perhaps to make comparison of results from multiple models. There is also a case for adoption of a comprehensive CSDMS (Peckham et al., 2013) type data and model management architecture in order to facilitate addition of other models and model components to the system. Some other useful additions include calibration modules, uncertainty parameters calculations etc. A wide array of input mechanisms, including support for user provided data source and a more comprehensive outputs such as 2-D map for each time step as provided by SWATShare (Rajib et al., 2016) would help even better in analysis. Lastly, support for snow-fed watersheds perhaps by coupling model with Utah Energy Balance (UEB) model (Tarboton and Luce, 1996) would also be very useful.

In summary, a web based solution for tackling data related problems in hydrologic modeling provides a number of benefits compared to conventional approach. More such works would make modeling simpler, add new user groups which would contribute in widening scope of hydrologic work. While there are number of limitations of the approach in general, and of the app presented here in particular, I believe the benefits outweigh them. Continued focus on similar works of providing modeling as service is required to help better understand hydrology.

6. CONCLUSIONS

The thesis presents a prototype web app that was developed to overcome the data related hurdles that exists in modeling. Specifically, the app helps overcome the problems of data discovery, retrieval, preprocessing, machine dependencies, and data storage. The work demonstrates the capability of such system to support modeling or in providing hydrologic data to end users. The web based system reduces complexities of modeling, is machine independent, and makes use of online computing and data storage. The improved reliability and usability not only save time of the hydrologists, they also open door for new user-groups to join the hydrologic discussion and expand the science.

The app developed made use of three major components- Tethys for frontend development, HydroDS for data processing, data storage and TOPKAPI modeling, and HydroShare for data and model storage. It supports modeling functionalities for the TOPKAPI model, input-files preparation for TOPNET, dataset access for terrain files, SSURGO based soil files and Daymet forcing files. Four use cases were adopted to demonstrate these functionalities. These functionalities are useful not only to experts, but also to beginner as it gives them access to model despite limited skillset.

The main finding of the work is not only the app itself, or the functionality it offers, but also is the proof of concept that a web based end-to-end hydrologic modeling web application is possible. With the development of platforms such as Tethys to support GUI development, HydroShare for model and data sharing, and web services such as HydroDS for data processing, a web based modeling system is not only achievable, but indeed becoming simpler than ever.

Although such web based solution for modeling have many benefits, they also have limitations. Most of the limitations arise from data sources, while others arise from the limitations of the model itself. Despite the limitations, there exist many benefits of web based modeling when used wisely. For further improving such systems, future work should focus on including a variety of data sources, offer modules for calibration and uncertainty assessment, include more models into the system, as well as improve input outputs system. With such improvements in the future works, hydrologic sciences will get closer than ever to fulfilment of the vision of Model as a Service and Web Science.

REFERENCES

- Abbott, M. B., J. C. Bathurst, J. A. Cunge, P. E. O'Connell and J. Rasmussen, (1986), "An introduction to the European Hydrological System—Système Hydrologique Européen, "SHE", 1: History and philosophy of a physically-based, distributed modelling system," Journal of Hydrology, 87(1-2): 45-59.
- Allen, R. G., W. O. Pruitt, J. L. Wright, T. A. Howell, F. Ventura, R. Snyder, D. Itenfisu, P. Steduto, J. Berengena and J. B. Yrisarry, (2006), "A recommendation on standardized surface resistance for hourly calculation of reference ET o by the FAO56 Penman-Monteith method," Agricultural Water Management, 81(1): 1-22.
- Ames, D., Z. Li, X. Qiao, D. G. Tarboton, R. Idaszak, J. Horsburgh, V. Merwade, B. Miles, N. Swain and R. Lineberger, (2015), "Web-Based Data Visualization and Analysis using HydroShare and the Open Source Tethys Platform," AGU Fall Meeting Abstracts.
- Ames, D. P., J. S. Horsburgh, Y. Cao, J. Kadlec, T. Whiteaker and D. Valentine, (2012), "HydroDesktop: Web services-based software for hydrologic data discovery, download, visualization, and analysis," Environmental Modelling & Software, 37: 146-156.
- Atkins, D., (2003), "Revolutionizing science and engineering through cyberinfrastructure: Report of the National Science Foundation blue-ribbon advisory panel on cyberinfrastructure."
- Bandaragoda, C., D. G. Tarboton and R. Woods, (2004), "Application of TOPNET in the distributed model intercomparison project," Journal of Hydrology, 298(1): 178-201.
- Bartholmes, J. and E. Todini, (2005), "Coupling meteorological and hydrological models for flood forecasting," Hydrology and Earth System Sciences Discussions, 9(4): 333-346.
- Bayer, M., (2012), "Sqlalchemy-the database toolkit for python," URL <http://www.sqlalchemy.org/>. Accessed on the 13th of November.
- Beaudette, D. E. and a. S. R. J. Skovlin, (2016), Soil Database Interface. <http://ncss-tech.github.io/AQP/>.
- Berners-Lee, T., W. Hall, J. Hendler and D. J. Weitzner, (2006), "Creating a Science of the Web," Science, 313(5788): 769-771.
- Best, B. D., P. N. Halpin, E. Fujioka, A. J. Read, S. S. Qian, L. J. Hazen and R. S. Schick, (2007), "Geospatial web services within a scientific workflow: Predicting marine mammal habitats in a dynamic environment," Ecological Informatics, 2(3): 210-223.
- Beven, K., (1989), "Changing ideas in hydrology—the case of physically-based models," Journal of Hydrology, 105(1-2): 157-172.
- Beven, K., M. Kirkby, N. Schofield and A. Tagg, (1984), "Testing a physically-based flood forecasting model (TOPMODEL) for three UK catchments," Journal of Hydrology, 69(1-4): 119-143.
- Beven, K. and M. J. Kirkby, (1979), "A physically based, variable contributing area model of basin hydrology/Un modèle à base physique de zone d'appel

- variable de l'hydrologie du bassin versant," Hydrological Sciences Journal, 24(1): 43-69.
- Choi, J.-Y., B. A. Engel and R. L. Farnsworth, (2005), "Web-based GIS and spatial decision support system for watershed management," Journal of Hydroinformatics, 7(3): 165-174.
- Ciarapica, L. and E. Todini, (2002), "TOPKAPI: A model for the representation of the rainfall - runoff process at different scales," Hydrological processes, 16(2): 207-229.
- Clark, M. P., D. E. Rupp, R. A. Woods, X. Zheng, R. P. Ibbitt, A. G. Slater, J. Schmidt and M. J. Uddstrom, (2008), "Hydrological data assimilation with the ensemble Kalman filter: Use of streamflow observations to update states in a distributed hydrological model," Advances in water resources, 31(10): 1309-1324.
- Coccia, G., C. Mazzetti, E. A. Ortiz and E. Todini, (2009), "Application of the TOPKAPI model within the DMIP 2 project," American Meteorological Society 23rd Conference on Hydrology, https://ams.confex.com/ams/89annual/techprogram/paper_149632.htm.
- Dash, P., (2016), Python client for accessing CI-WATER data and computational services, https://github.com/CI-WATER/hydrogate_python_client, accessed 02/20/2017.
- Dawson, C. W., R. J. Abrahart and L. M. See, (2007), "HydroTest: a web-based toolbox of evaluation metrics for the standardised assessment of hydrological forecasts," Environmental Modelling & Software, 22(7): 1034-1052.
- Geller, G. N. and F. Melton, (2008), "Looking forward: Applying an ecological model web to assess impacts of climate change," Biodiversity, 9(3-4): 79-83.
- Geller, G. N. and W. Turner, (2007), "The model web: a concept for ecological forecasting," Geoscience and Remote Sensing Symposium, 2007. IGARSS 2007. IEEE International, IEEE, 2469-2472, 1424412110.
- Goodall, J. L., J. S. Horsburgh, T. L. Whiteaker, D. R. Maidment and I. Zaslavsky, (2008), "A first approach to web services for the National Water Information System," Environmental Modelling & Software, 23(4): 404-411.
- Granell, C., L. Díaz and M. Gould, (2010), "Service-oriented applications for environmental models: Reusable geospatial services," Environmental Modelling & Software, 25(2): 182-198.
- Horsburgh, J. S., D. G. Tarboton, M. Piasecki, D. R. Maidment, I. Zaslavsky, D. Valentine and T. Whitenack, (2009), "An integrated system for publishing environmental observations data," Environmental Modelling & Software, 24(8): 879-888.
- Ibbitt, R. and R. Woods, (2004), "Re-scaling the topographic index to improve the representation of physical processes in catchment models," Journal of Hydrology, 293(1): 205-218.
- Jeong, S., Y. Liang and X. Liang, (2006), "Design of an integrated data retrieval, analysis, and visualization system: application in the hydrology domain," Environmental Modelling & Software, 21(12): 1722-1740.

- Kalyanapu, A. J., S. J. Burian and T. N. McPherson, (2010), "Effect of land use-based surface roughness on hydrologic model output," Journal of Spatial Hydrology, 9(2).
- Leonard, L. and C. J. Duffy, (2013), "Essential Terrestrial Variable data workflows for distributed water resources modeling," Environmental Modelling & Software, 50: 85-96.
- Lim, K. and B. A. Engel, (2000), "Evolution and evaluation of nutrient enabled NAPRA WWW decision support system with GLEAMS 3.0," Evolution and evaluation of nutrient enabled NAPRA WWW decision support system with GLEAMS 3.0.: 1-8.
- Liu, Z., M. L. Martina and E. Todini, (2005), "Flood forecasting using a fully distributed model: application of the TOPKAPI model to the Upper Xixian Catchment," Hydrology and Earth System Sciences Discussions, 9(4): 347-364.
- Liu, Z. and E. Todini, (2002), "Towards a comprehensive physically-based rainfall-runoff model," Hydrology and Earth System Sciences Discussions, 6(5): 859-881.
- Loucks, D. P., J. Kindler and K. Fedra, (1985), "Interactive water resources modeling and model use: an overview," Water Resources Research, 21(2): 95-102.
- McKee, L., (2000), Catch the Internet Wave. <http://www.opengis.org>.
- McMillan, H., J. Freer, F. Pappenberger, T. Krueger and M. Clark, (2010), "Impacts of uncertain river flow data on rainfall - runoff model calibration and discharge predictions," Hydrological processes, 24(10): 1270-1284.
- Miles, B. and L. E. Band, (2015), "Ecohydrology Models without Borders?," International Symposium on Environmental Software Systems, Springer, 311-320.
- Miller, S. N., D. J. Semmens, D. C. Goodrich, M. Hernandez, R. C. Miller, W. G. Kepner and D. P. Guertin, (2007), "The automated geospatial watershed assessment tool," Environmental Modelling & Software, 22(3): 365-377.
- Mineter, M. J., C. Jarvis and S. Dowers, (2003), "From stand-alone programs towards grid-aware services and components: a case study in agricultural modelling with interpolated climate data," Environmental Modelling & Software, 18(4): 379-391.
- Pandey, S., R. Gunn, K. J. Lim, B. Engel and J. Harbor, (2000), "Developing a Web-enabled tool to assess long-term hydrologic impacts of landuse change: Information technology issues and a case study," URISA-WASHINGTON DC, 12(4): 5-22.
- Pecar-Ilic, J. and I. Ruzic, (2006), "Application of GIS and Web technologies for Danube waterway data management in Croatia," Environmental Modelling & Software, 21(11): 1562-1571.
- Peckham, S. D., E. W. Hutton and B. Norris, (2013), "A component-based approach to integrated modeling in the geosciences: The design of CSDMS," Computers & Geosciences, 53: 3-12.
- Pellicciotti, F., C. Buergi, W. W. Immerzeel, M. Konz and A. B. Shrestha, (2012), "Challenges and uncertainties in hydrological modeling of remote Hindu

- Kush–Karakoram–Himalayan (HKH) basins: suggestions for calibration strategies," Mountain Research and Development, 32(1): 39-50.
- Rajasekar, A., R. Moore, C.-y. Hou, C. A. Lee, R. Marciano, A. de Torcy, M. Wan, W. Schroeder, S.-Y. Chen and L. Gilbert, (2010), "iRODS Primer: integrated rule-oriented data system," Synthesis Lectures on Information Concepts, Retrieval, and Services, 2(1): 1-143.
- Rajib, M. A., V. Merwade, I. L. Kim, L. Zhao, C. Song and S. Zhe, (2016), "SWATShare–A web platform for collaborative research and education through online sharing, simulation and visualization of SWAT models," Environmental Modelling & Software, 75: 498-512.
- Rawls, W. J., D. Brakensiek and K. Saxton, (1982), "Estimation of soil water properties," Transactions of the ASAE, 25(5): 1316-1320.
- Richards, M., (2015), Python package for calculating reference/potential evapotranspiration (ET_o), GitHub, <https://github.com/woodcrafty/PyETo>, accessed 2017/11/13.
- Roman, D., S. Schade, A. Berre, N. R. Bodsberg and J. Langlois, (2009), "Model as a service (MaaS)," AGILE Workshop: Grid Technologies for Geospatial Applications, Hannover, Germany.
- Sazib, N. S., (2016), "Physically Based Modelling of the Impacts of Climate Change on Streamflow Regime," Doctoral dissertation, Utah State University Thesis, <https://digitalcommons.usu.edu/cgi/viewcontent.cgi?article=6110&context=etd>.
- Shea, J., S. Ragettli, W. Immerzeel, F. Pellicciotti, E. Miles, J. Steiner, P. Buri and P. Kraaijenbrink, (2015), "After the Earthquake: Impacts of Seismic Snow and Ice Redistribution in Langtang Valley, Nepal, on Glacier Mass Balances and Hydrological Regimes," AGU Fall Meeting Abstracts.
- Shneiderman, B., (2007), "Web science: a provocative invitation to computer science," Communications of the ACM, 50(6): 25-27.
- Sinclair, S. and G. Pegram, (2010), "A comparison of ASCAT and modelled soil moisture over South Africa, using TOPKAPI in land surface mode," Hydrology and Earth System Sciences, 14(4): 613.
- Sinclair, S. and G. Pegram, (2013), "A sensitivity assessment of the TOPKAPI model with an added infiltration module," Journal of Hydrology, 479: 100-112.
- Singh, V. P. and D. K. Frevert, (2002), Mathematical models of large watershed hydrology, Water Resources Publication.
- Soh, L. K., J. Zhang and A. Samal, (2006), "A Task - Based Approach to User Interface Design for a Web - Based Hydrologic Information Systems," Transactions in GIS, 10(3): 417-449.
- Swain, N. R., S. D. Christensen, A. D. Snow, H. Dolder, G. Espinoza-Dávalos, E. Goharian, N. L. Jones, E. J. Nelson, D. P. Ames and S. J. Burian, (2016), "A new open source platform for lowering the barrier for environmental web app development," Environmental Modelling & Software, 85: 11-26.
- Tarboton, D., (2016), HydroShare and Model Sharing: Presentation to IWRSS Model Registry Team, Nov 8, 2016.
<http://www.hydroshare.org/resource/26d30f89e31f481bb63a4e089dfa1340>.

- Tarboton, D. G., (2005), "Terrain analysis using digital elevation models (TauDEM)," Utah State University, Logan.
- Tarboton, D. G., (2007a), Surface Water Quantity Model Development and Calibration. Task 4.1 report:
<http://wrialproject.whatcomcounty.org/uploads/USU/2008/Task4.1-SWQNcalibration.pdf>.
- Tarboton, D. G., (2007b), Validation of Surface Water Quantity Model through Analyses of Scenarios. Task 4.1 report:
<http://wrialproject.whatcomcounty.org/uploads/USU/2008/Task4.2-SWQNvalidation.pdf>.
- Tarboton, D. G., R. Idaszak, J. Horsburgh, D. Ames, J. Goodall, L. Band, V. Merwade, A. Couch, J. Arrigo and R. Hooper, (2013), "HydroShare: an online, collaborative environment for the sharing of hydrologic data and models," AGU Fall Meeting Abstracts, 1, 1510.
- Tarboton, D. G., R. Idaszak, J. S. Horsburgh, J. Heard, D. Ames, J. L. Goodall, L. Band, V. Merwade, A. Couch and J. Arrigo, (2014), "HydroShare: advancing collaboration through hydrologic data and model sharing."
- Tarboton, D. G. and C. H. Luce, (1996), Utah energy balance snow accumulation and melt model (UEB), Utah Water Research Laboratory.
- Taylor, K., G. Walker and D. Abel, (1999), "A framework for model integration in spatial decision support systems," International Journal of Geographical Information Science, 13(6): 533-555.
- Teng, J., J. Vaze, N. K. Tuteja and J. C. Gallant, (2008), "A GIS - Based Tool for Spatial and Distributed Hydrological Modelling: CLASS Spatial Analyst," Transactions in GIS, 12(2): 209-225.
- Todini, E. and L. Ciarapica, (2001), The TOPKAPI model. Mathematical Models of Large Watershed Hydrology, Chapter 12, edited by Singh. VP et al.
- Vischel, T., G. Pegram, S. Sinclair and M. Parak, (2008a), "Implementation of the TOPKAPI model in South Africa: Initial results from the Liebenbergsvlei catchment," Water Sa, 34(3): 331-342.
- Vischel, T., G. Pegram, S. Sinclair, W. Wagner and A. Bartsch, (2008b), "Comparison of soil moisture fields estimated by catchment modelling and remote sensing: a case study in South Africa," Hydrology and Earth System Sciences Discussions, 12(3): 751-767.
- Walker, J. D. and S. C. Chapra, (2014), "A client-side web application for interactive environmental simulation modeling," Environmental Modelling & Software, 55: 49-60.
- White, T., (2012), Hadoop: The definitive guide, " O'Reilly Media, Inc."
- Wieczorek, M. E., (2014), Area- and Depth-Weighted Averages of Selected SSURGO Variables for the Conterminous United States and District of Columbia. https://water.usgs.gov/lookup/getspatial?ds866_ssurgo_variables.
- Zaslavsky, I., D. Valentine and T. Whiteaker, (2007), "CUAHSI WaterML," Open Geospatial Consortium Discussion Paper OGC.

APPENDICES

Appendix A. Abbreviation and Glossary of Terms

Following abbreviation, and terminology are used in this paper:

API	Application Programming Interface
CI	Cyber Infrastructure
CONUS	Continental United States
CSS	Cascading Style Sheets
DEM	Digital Elevation Model
EPSG	European Petroleum Survey Group
ESRI	Environmental System Research Institute
ET	Evapotranspiration
FAO	Food and Agricultural Organization
FOSS	Free and Open Source Software
GeoJSON	Geographic JavaScript Object Notation
GIS	Geographic Information System
gSSURGO	gridded Soil Survey Geographical database
GUI	Graphical User Interface
HDF	Hierarchical Data Format
HTML	Hyper Text Markup Language
JSON	JavaScript Object Notation
LUT	Look-up Table
NED	National Elevation Dataset
NetCDF	Network Common Data Form
NLCD	National Land Cover Dataset
NRCS	Natural Resources Conservation Service
NSE	Nash–Sutcliffe Efficiency
NSF	National Science Foundation
ORM	Object-relational Mapping
SDA	Soil Data Access
SQL	Structured Query Language
SRTM	Shuttle Radar Topography Mission
SSURGO	Soil Survey Geographical database
STATSGO	State Soil Geographic Database
TIFF	Tagged Image Format File
UEB	Utah Energy Balance
URL	Universal Resource Locator
US	United States
USGS	United Nations Geographic Survey

Appendix B. SSURGO Dataset and Calculations

SSURGO divides a region into multiple polygonal shape based on similar soil properties- called map-units. Each of those map units contain different components of soil on it. Total percentage of a particular component of soil present in each map units is available, however, components lack spatial information. Hence, soil properties can only be analyzed per map unit.

The properties for each component needs to be averaged out based on their presence percentage in corresponding map units. Most of the soil properties (for example soil type, porosity etc.) are associated with the soil layers (soil horizon in SSURGO term), not components. Thus, a representative value for each component needs to be calculated by taking weighted average based on thickness of the soil layers present. Figure B.1 illustrates the relationship between soil horizons and components for one map unit. A calculation example for finding k_{sat} (saturated hydraulic conductivity) value for one map-unit (map unit 1) is explained in the Figure. Notice in the upper portion of the figure where k_{sat} value was calculated by taking weighted average based on height of the soil layer. The value obtained, 2.95 is value of one of four components present in the map unit we are concerned. To calculate k_{sat} representing all of map-unit 1, another weighted average based on component percentage will need to be done, as shown in the lower part of the Figure B.1. A script was written in R to perform similar calculations as shown in Figure to obtain soil property values for each map units.

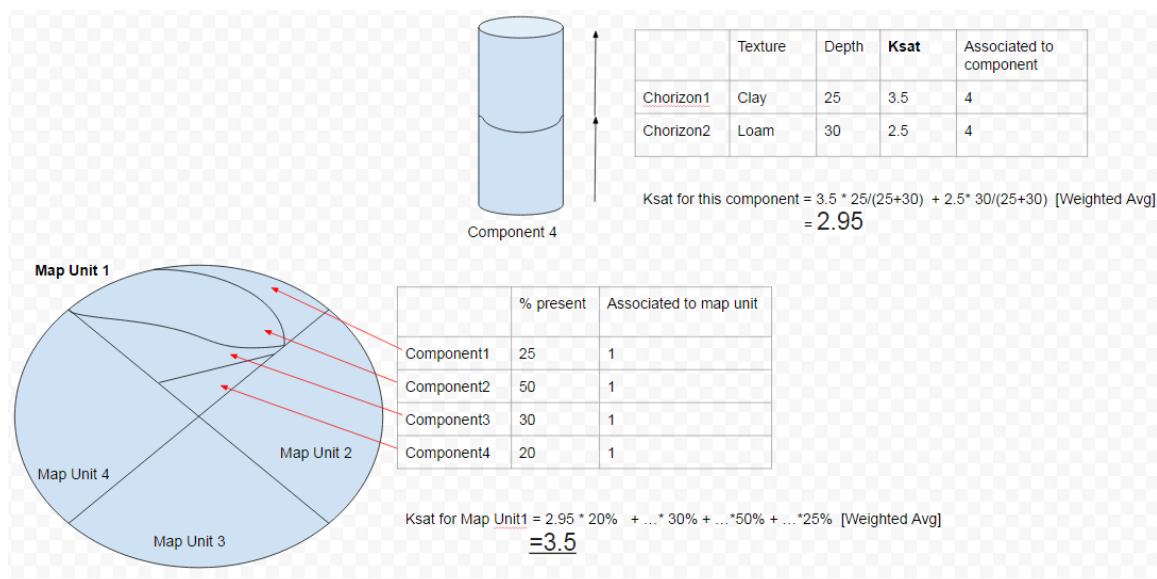


Figure B.1. Schematic diagram of SSURGO representation of soil properties, and their calculations. Saturated Hydraulic Conductivity value (ksat) for the two example layers of a component (with name component4) was weighted-averaged based on thickness of soil horizons to obtain one value of 2.95 for it. Four similar component values were weighted-averaged based on component percentage to obtain one value of 3.5 for Map unit 1.

Many soil properties values are either unavailable or incomplete in the SSURGO database. To calculate such properties, texture class information for each soil horizons was used to estimate soil properties for that texture class using Rawls lookup table. The process of obtaining soil texture class for each soil horizon is also tricky. Each texture class is mapped with corresponding soil horizon based on 'Chorizon Texture Group' table, hence texture class for each soil horizon was found out by first joining 'texture' table with 'texture group' table, which was again joint to 'horizon' table. The relationship may be better explained by the entity relationship diagram below (Figure B.2).

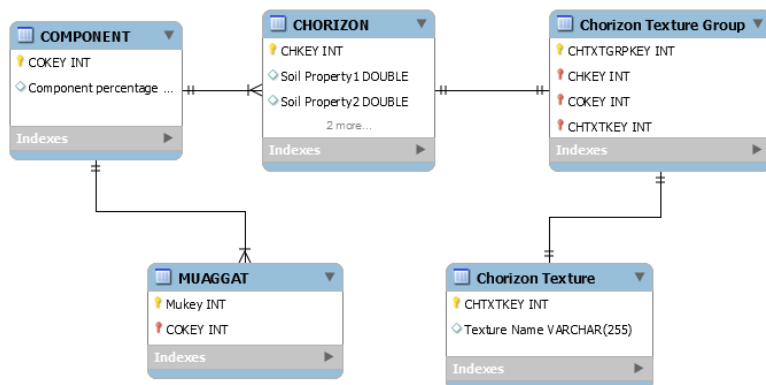
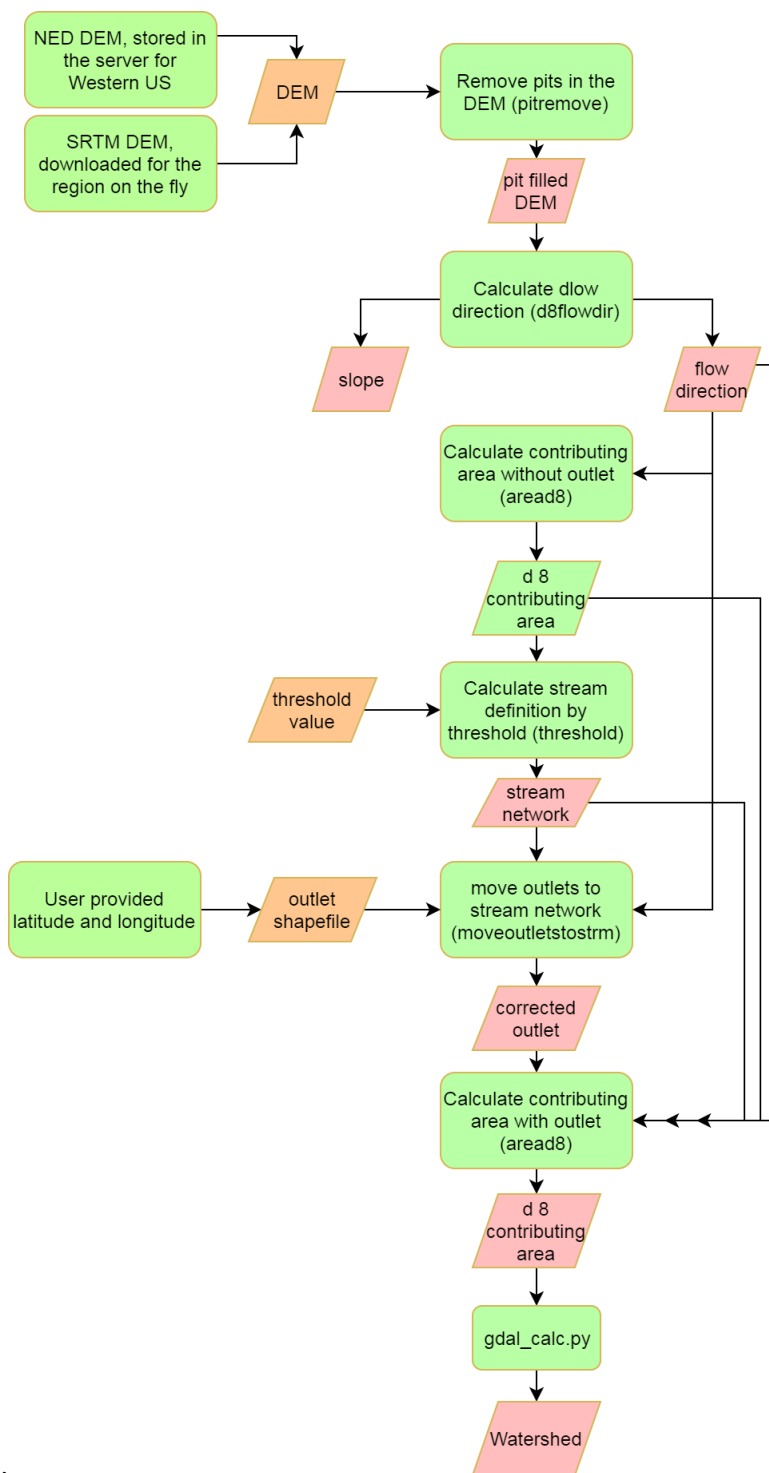


Figure B.2. Entity relationship diagram of SSURGO tables used in the project.

Appendix C. Terrain Analysis Using TauDEM

The terrain analysis workflow using TauDEM tools is shown in the Figure C.1

Terrain analysis involve delineation of stream network from DEM, identification of terrain driven surface flowpath, and identification of watershed area draining into an outlet etc. First, a DEM (either saved in servers or downloaded using data service) is extracted for the user defined domain, projected and resampled to the user defined cell size. Pits in the DEM are filled. Flow direction, which identifies the direction (left, right, top, bottom, or diagonal) where the cell flows to based on the slope of the terrain, and flow accumulation rasters, which define total number of cell that drains to each cell are computed. Using these rasters, and user input “stream threshold” value, the stream network is obtained from the DEM. Stream threshold value is total number of cells required to drain to a cell to be considered as a stream cell. The user input outlet information may not be accurate, i.e. the outlet may not fall exactly on the stream. For such cases, TauDEM function ‘moveoutlettostream’ is used to create a new shapefile containing a point for outlet in the stream. Finally, all the cells flowing into the outlet are grouped using GDAL to create a watershed map. The watershed raster is converted to ESRI polygon shapefile and GeoJSON files. Hence, the final output of the analysis include a DEM, flow direction map, stream network map, slope map, watershed coverage



maps.

Figure C.1. Workflow employed for terrain analysis using TauDEM. Green square boxes represent process, pink trapezoid represent outputs, while brown trapezoid represents inputs.

Appendix D. HydroDS Functions

The data preprocessing and modeling functionalities were accomplished by functions (Python and R) in HydroDS that utilized packages such as GDAL, NCO and TauDEM. This section briefly describes the existing HydroDS functions that were used, as well as the new functions created during the course of the project.

Existing Functions

The existing HydroDS functions used by the app are listed (Table D.1).

Table D.1. Existing HydroDS services used

Function name as webservice	Description
subetrastertobbox	Extracts the DEM and NLCD grid from nationwide sources for the model domain.
projectraster	Projects the raster files to a projected coordinate system
resampleraster	Resamples the raster to a user desired resolution based on standard resampling techniques
createoutletshapefile	Creates outlet shapefile from the user input coordinate of outlet, to be used as input to function to delineate watershed
NetCDFrenamevariable	Renames NetCDF files
concatenateNetCDF	Concatenates climate NetCDF files when the time span for the output file spans multiple years
subsetNetCDFbytime	Subsets offline NetCDF climate files to a specified time
subsetNetCDF - toreference	Subsets offline NetCDF climate files to a specific raster
projectsubset – resampleNetCDF - toreferenceNetCDF	Projects, subsets, and resamples a NetCDF file based on reference NetCDF file's projection, cell size
projectshapefileepsg	Projects a ESRI shapefile based on a given EPSG code
computerasteraspect	Computes aspect for a given raster

Created Functions

The functions created during the course of the project are listed (Table D.2). While some of the functions are generic (applicable to most hydrologic models), a few are only applicable to TOPKAPI.

Table D.2. New HydroDS services created during the project

Functions as webservicea	Description
bboxfromshp	Gives bounding box coordinates (in WGS) from polygon shapefile
bboxfromtiff	Gives bounding box coordinates (in WGS) from GeoTIFF file
outletxyfromshp	Gives latitude and longitude (in WGS) from point shapefile
downloadglobaldem	Downloads SRTM DEM (30m or larger) for any global area
getrasterdetail	Gives raster metadata such as cell size, cell counts
rastersubset2	Subsets offline NED-DEM if bounding box located in western US, else download SRTM dem
reclassifyraster withlookuptable	Reclassifies raster using lookup table
delineatewatershed togetcompleterasterset	Perform TauDEM assisted delineation watershed to create stream network, d-8 flow direction raster, raster, slope raster, and the watershed draining to the given outlet
downloadsoil dataforpytopkapi	Queries NRCS-SDA servers and create rasters defining soil properties such as saturated hydraulic conductivity, porosity, pore-size distribution, saturated soil moisture, residual soil moisture, soil depth,
calculatereferenceet fromdaymet	Extracts Daymet climate variables (max temperature, min temperature, vapor pressure, incoming solar radiation), and DEM raster, and use them to calculate short crop reference potential evapotranspiration using FAO guidelines
downloadand resampleusgs	Fetches the observed discharge for the time period at the given USGS stream gage located close to the outlet of the watershed
createpytopkapi forcingfile	Converts forcing files: rainfall and evapotranspiration from NetCDF to the PyTOPKAPI accepted HDF5 format
create_config_files_ xxx	Creates PyTOPKAPI configuration files (many such functions starting with similar names to create many such files) based on inputs
runpytopkapi	Runs a PyTOPKAPI model once the input-files are prepared
loadpytopkapi	Loads a previously prepared PyTOPKAPI model given HydroShare resource ID for the model-instance location
modifypytopkapi	Modifies a loaded PyTOPKAPI model instance
runtopnet	Creates input files from TOPNET model
downloadgeospatialfiles	Gives terrain files, soil files or climate/forcing files for a domain

Appendix E. Output JSON Summary File

A screenshot of the JSON string contained in the app info text file (saved in HydroShare as **app_info.txt**) that keeps the run information. The file not only keeps the input records and metadata information, but also contains output records like time series values.

```
{
  "user_name": "prasanna_310",
  "simulation_name": "Example_run",
  "simulation_start_date": "10/01/2010",
  "cell_size": 500.0,
  "USGS_gage": "11124500",
  "units": {
    "timeseries_list": "[year, month, day, hour, min, float-value]",
    "watershed_area": "square meters",
    .
    .
    .
  },
  "timestep": 24,
  "watershed_area": 72000000.0,
  "threshold": 5,
  "simulation_end_date": "10/03/2010",
  "observed_discharge": [ [ 2010,10,2,0,0, 0.02 ], ...],
  "ppt": [...],
  "runs": [
    {
      "run_id": 1,
      "et_a": [...],
      "calib_parameter": {...},
      "errors": {...},
      "simulated_discharge": [...],
      "vo": [...],
      "vs": [...],
      "numeric_param": {...},
      "vc": [...],
    }
    {
      "run_id": 2,
      "et_a": [...],
      "calib_parameter": {...},
      "errors": {...},
      "simulated_discharge": [...],
      "vo": [...],
      "vs": [...],
      "numeric_param": {...},
      "vc": [...],
    }
  ]
}
```

Figure E.1. JSON string contained in the app info text file that keeps the run information. The record for new runs get added to the records in the “run” key as dictionary:

Appendix F. App Tutorial: Prepare TOPKAPI Model

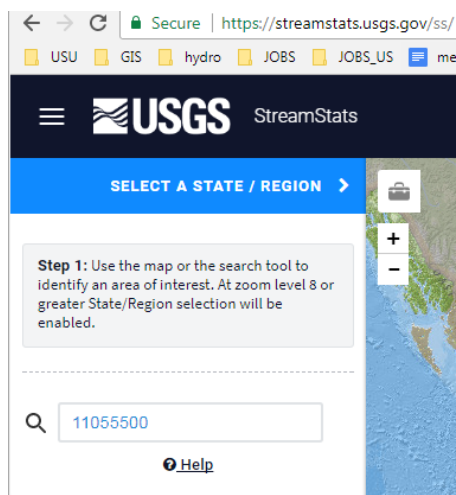
This tutorial provides step by step guidance for using the HydroTop app to prepare input-files for TOPNET. The steps given correspond to the Plunge River, the alternative use case mentioned in the text to demonstrate app's TOPKAPI modeling functionality.

Log on to the app's home page (<http://appsdev.hydroshare.org/apps/hydrotop>). If you are prompted to enter your HydroShare account, and to authorize the app to access your HydroShare account, please do so. As of this writing, the app is hosted by the HydroShare's development app portal, but this could change in the future.

We are interested in modeling a watershed called Plunge that drains to the USGS gage 11055500 in south west California. We will need to know the position of the outlet, and the domain containing the watershed. Assuming we have the required information, lets proceed. The next section shows the procedure to get the model domain accurately using streamstats, which is the recommended method to define the model domain. You can skip the following section (*Get bounding box GeoJSON from Streamstat*) and go to *App Input* section if you already have a file describing the bounding box, or if you only want the App tutorial.

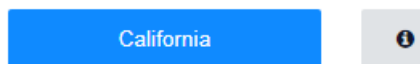
Get bounding box GeoJSON from Streamstat

There are three ways to enter the domain, drawing on the map, explicitly typing them, or uploading supported file. In this tutorial, we will use a combination of the two. First, we will get a GeoJSON file for our watershed to accurately describe our model domain. For this, go to <https://streamstats.usgs.gov/ss/>, and enter the USGS gage id 11055500 in the search box field to the left of the screen.



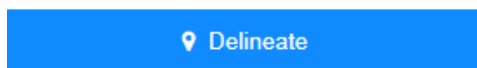
A button to confirm the location is California pops up. Click on it to confirm.

Click to select a State or Regional Study Area

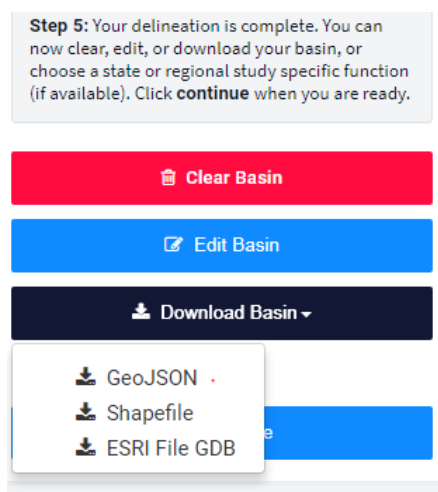


We want to delineate watershed for our USGS gage, so click on Delineate button to do so.

Step 2: Click the 'Delineate' button to activate the delineation tool



Now click on the stream location on the map, very close to the USGS gage location.



App Inputs

Switch back to the app's homepage. In the app navigation section to the top left of screen, click and expand the tab '**1. General Inputs**', and type in the simulation name, and start and end date of the simulation. In this example, we will do 2010 water year.

1. General Inputs

Simulation name

Plunge_2010_demo

USGS gage nearby

11055500

Start Date

10-01-2010

End Date

10-01-2011

2. The domain

3. Additional Inputs

Expand '**The Domain**' section, and type in the latitude and longitude of the outlet, which is the location of USGS gage, as 34.12128 and -117.141284 respectively (Figure ...,

a). This should bring the outlet marker on the map to our point of interest. On the same section, check the “**Upload GeoJSON file for watershed**” button → **upload the GeoJSON file** you just downloaded → Click ‘**upload file**’ button.

2. The domain

☐ Specify long lat, or draw on map

OUTLET

Longitude	Latitude
-117.141284	34.12128

BOUNDING BOX

North Y

34.21972

West X

-117.16

East X

-117.084

South Y

34.11604

☐ Upload shp files or TIF

☒ Upload geojson files for watersheds

Choose File **Plunge.geojson**

Click here to Upload

Latitude: 34.0528 Longitude: -117.1472

The map section should be updated with the watershed map and the bounding box surrounding the watershed. Also, the coordinates for the bounding box at the values at the navigation pane also should update accordingly.

Now, head over to the navigation pane, click and expand the tab the third tab-‘**3. Additional Inputs**’, complete the form as shown in the figure below.

3. Additional Inputs

Cell size in meters

100

Stream threshold in square km

5

Choose an action

Prepare TOPKAPI model

Initial saturation in soil cells (in %)

30

Initial volume of water in overland cells (in m3)

3

Initial flow of water in channel cells (in m3/s)

0.1

Send Request

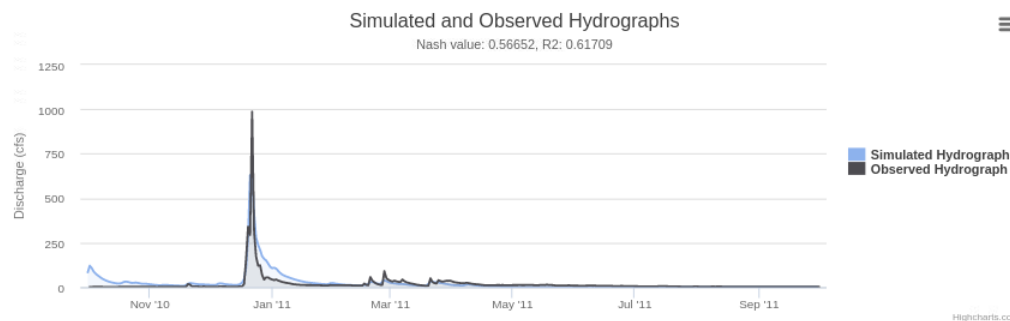
This is all the inputs required to create a TOPKAPI model. With the inputs completed, go ahead and click ‘**Send Request**’ button. It will be few minutes for the app to do its work and return result. So make sure you don’t close the page. If you accidentally closed it, that will be fine too. The model-instance will be saved in your HydroShare account so you will not lose the progress.

Results

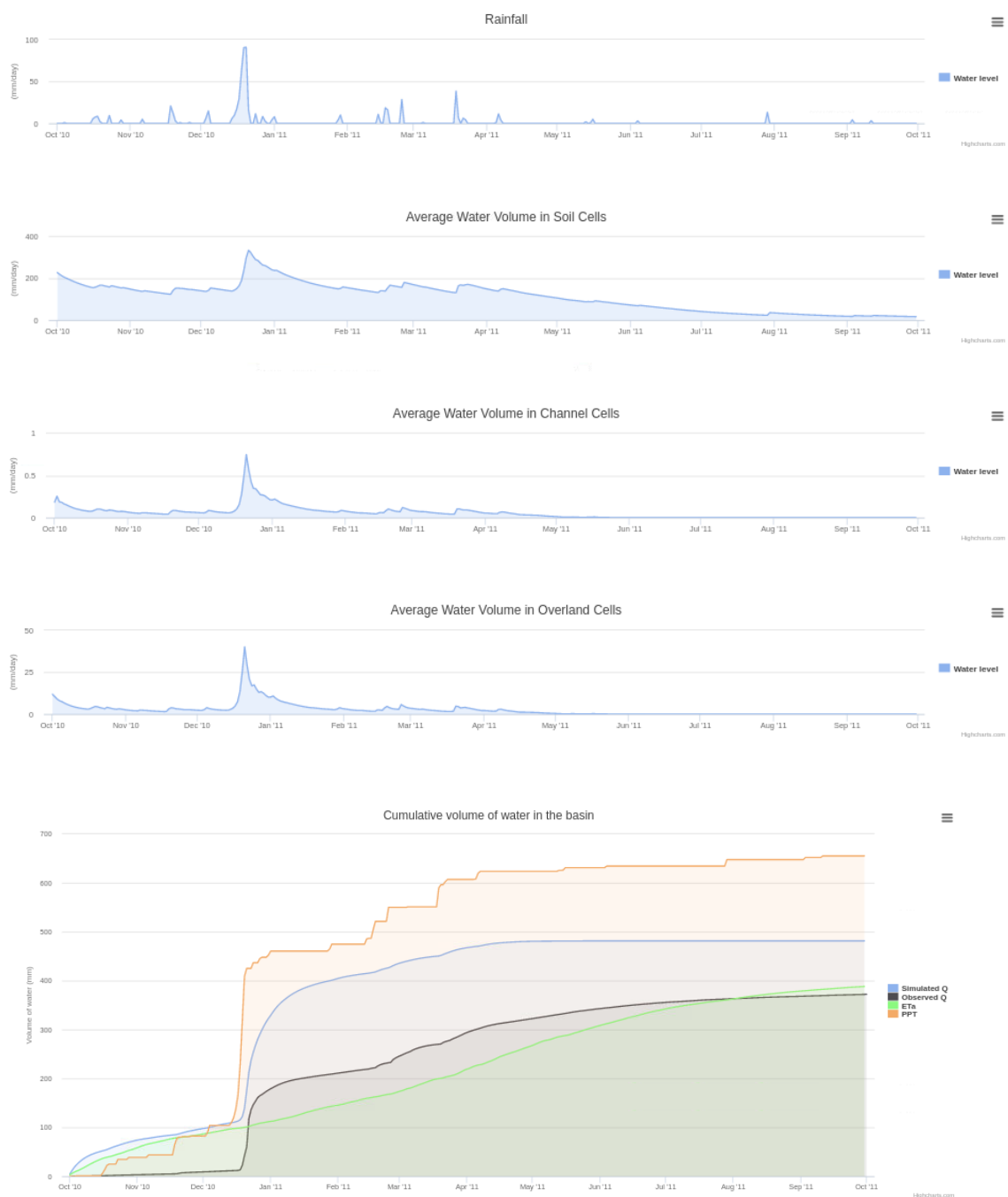
After a few minutes, you should get results (as shown in the figure below), which are the time series plots of the simulated and observed hydrograph. The plot also mentions a NSE value of 0.57, and an R^2 value of 0.62 for the two hydrographs.

PyTOPKAPI simulation- Plunge_demo ran successfully!

Results saved in HydroShare with resource ID: [0c6519de865c4ce78c28c9fae54b82c8](#) Here is the simulated discharge for the outlet chosen



There are also other time series plots displayed as shown in the figure below, like rainfall (Fig (a)), volume of water in soil (Fig (b)), channel (Fig (c)) and overland (Fig (d)) cells and actual ET (Fig (e)). Additionally, a cumulative graphs of volume of water in the basin (Fig (f)).



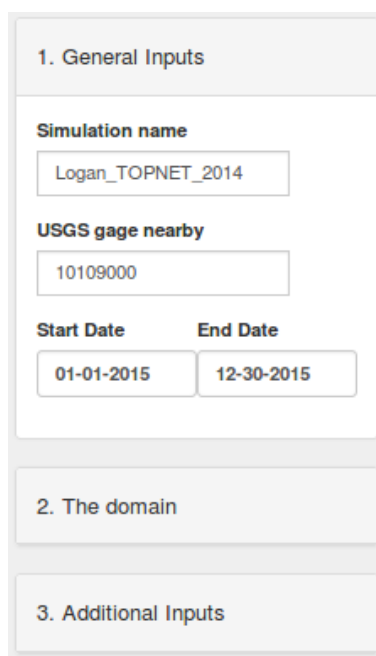
The initial values used for the model are displayed in the left section of the app, which is shown in the figure below:

<u>Multiplicative factors</u>	
Soil depth across all model cells	Saturated hydraulic conductivity
<input type="text" value="1.0"/>	<input type="text" value="1.0"/>
Manning's n for overland	Manning's n for channel
<input type="text" value="1.0"/>	<input type="text" value="1.0"/>
Soil saturation	
<input type="text" value="1.0"/>	
<u>Initial Parameters</u>	
Soil cell's saturation %	Water volume in Overland cells (m3)
<input type="text" value="30.0"/>	<input type="text" value="3.0"/>
Flow in channel cells (m3/s)	Crop coefficient across all model cells
<input type="text" value="0.1"/>	<input type="text" value="1.0"/>
<input type="button" value="Submit"/>	

Appendix G. App Tutorial: Prepare TOPNET Input Files

This tutorial provides step by step guidance for using the HydroTop app to prepare input-files for TOPNET. The steps given correspond to the Logan River use case for TOPNET input files creation described in the text.

To start, log on to <http://appsdev.hydroshare.org/apps/hydrotop>, the app's home page. In the *app navigation* section to the top left of screen, click and expand the tab '**1. General Inputs**', and type in the simulation name, and start and end date of the simulation. In this example, we will do 2015 calendar year.



The screenshot shows the '1. General Inputs' section of the HydroTop app. It contains the following fields and values:

- Simulation name:** Logan_TOPNET_2014
- USGS gage nearby:** 10109000
- Start Date:** 01-01-2015
- End Date:** 12-30-2015

Below the 'General Inputs' section are two more sections: '2. The domain' and '3. Additional Inputs', which are currently collapsed.

Expand '**The Domain**' section and click '**Specify long lat**' radio button. Type in the latitude and longitude of the outlet, which is the location of USGS gage 10109000, as 41.7436 and -111.7836 respectively. This should bring the outlet marker on the map to our point of interest. On the same section, enter the bounding box coordinates for North Y, East X, South Y and West X as 42.12, -111.44, 41.68, and -111.83 respectively.


2. The domain

☐ Specify long lat, or draw on map

OUTLET
Longitude Latitude
-111.7836 41.7436

BOUNDING BOX
North Y
42.12
West X East X
-111.83 -111.44
South Y
41.68

☐ Upload shp files or TIF
☐ Upload geojson files for watershed



Latitude: 41.6780 Longitude: -111.7886

The map section should be updated with the watershed map and the bounding box surrounding the watershed. Now, head over to the navigation pane, click and expand the tab the third tab-**‘3. Additional Inputs’** → **‘Prepare TOPNET input-files’**, and complete the expanded form for TOPNET inputs as shown in the figure below.

3. Additional Inputs

Cell size in meters
30

Stream threshold in square km
25

Choose an action
Prepare TOPNET input-files

Stream threshold
100

pk_min_threshold
500

pk_max_threshold
50000

pk_num_threshold
12

Send Request

This is all the inputs required to create input files for TOPNET model. With the inputs completed, go ahead and click ‘**Send Request**’ button. It will be few minutes for the app to do its work and return result.

Results

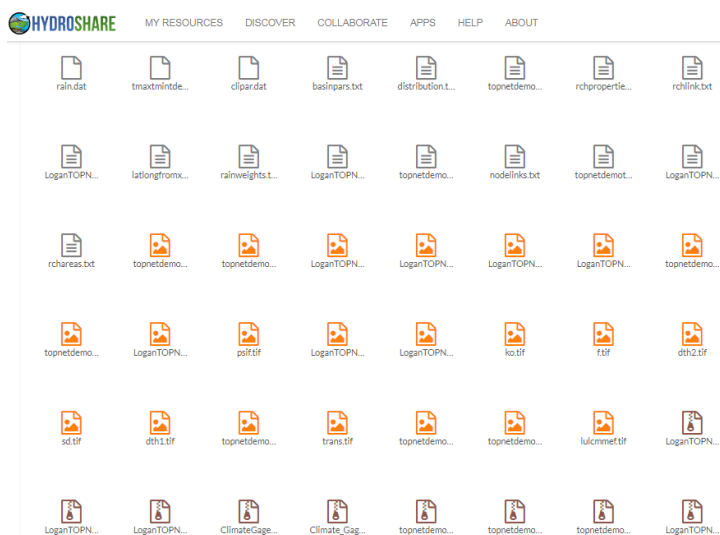
After a few minutes, you should get the prompt (as shown in the figure below), saying the files were created successfully. This page will have a temporary link to the zipped file that contains all the files for you to download immediately. Also, you will have a link to HydroShare where the file is permanently stored. The results of this analysis can be found in HydroShare, and has been made public and available at <https://www.hydroshare.org/resource/955d72a9b00141548aca582c493ee140/>

Congratulations! Files were created successfully!

The created geospatial and forcing files can be accessed temporarily by using this link: [Download zip here](#)

The files is also saved permanently in HydroShare, with the ID [955d72a9b00141548aca582c493ee140](#)

Snippet of files shaved in HydroShare is a shown in the figure below.

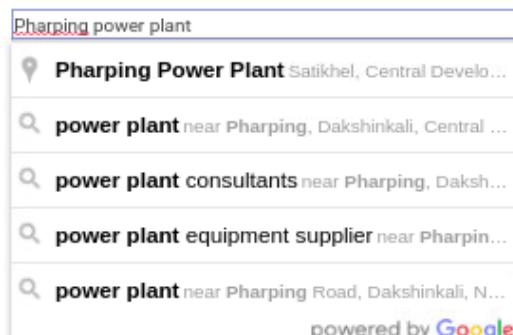



Appendix H. App Tutorial: Perform Terrain Analysis

This tutorial provides step by step guidance for using the HydroTop app to perform terrain analysis. The steps given correspond to the terrain analysis for the Bagmati River in Kathmandu, Nepal use case.

To start, log on to the app's home page (<http://appsdev.hydroshare.org/apps/hydrotop>). The outlet location, and the extent of the area that includes the watershed are required as inputs. Assuming we have the required information, lets proceed.

To quickly go to our desired location, type in 'Pharping Power Plant' in the search box located at the top-left of the map element.



Now, click on this icon at the top-center of the map element,  and draw the outlet point on the Bagmati River.



Now zoom out until you see 5km in the scale at the bottom-right of the map



. Draw a rectangle that completely surrounds the valley using the ‘rectangle draw’ toolbar from the top- center of the screen.



In the app navigation section, type in the name of the simulation as ‘BagmatiRiver-Kathmandu’ or any other name that you would want to give. This name will be used in HydroShare to name the resources. You can ignore the other inputs in the tabs ‘1. General Inputs’ and ‘2. Domain’. Now head over to the additional input tab and fill the form as shown in the figure below. Because this area is outside US, only ‘downloading terrain files’ option will work, which should be checked by default when you chose ‘Download geospatial files’ in the ‘Chose an action’ menu.

This is all the inputs required to create perform terrain analysis. With the inputs completed, go ahead and click ‘Send Request’ button. It will be few minutes for the app to do its work and return result. So make sure you don’t close the page. Again, if you accidentally closed it, that will be fine too. The files will be saved in your HydroShare account so you will not lose the progress.

Results

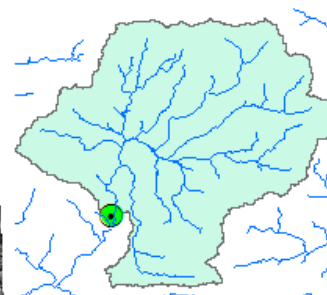
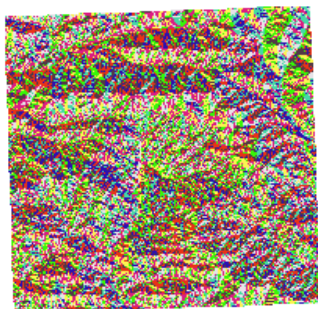
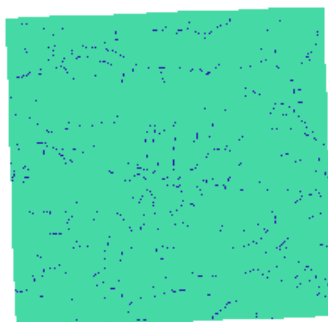
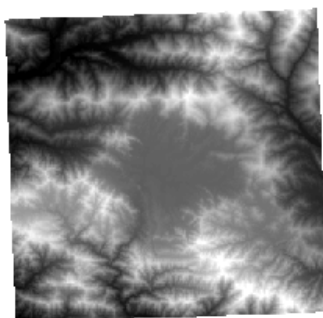
After a few minutes, you should get results page displayed, which will have a temporary link to the zipped file that contains all the files for you to download immediately. Also, you will have a link to HydroShare where the file is permanently stored. The results of this analysis can be found in HydroShare, and has been made public. If you want to

check it out, here is the link: <https://www.hydroshare.org/resource/ebda23b9e6054f5d8e6dc1ddb1594d26>

Congratulations! Files were created successfully!

The created geospatial and forcing files can be accessed temporarily by using this link: [Download zip here](#)

The files is also saved permanantly in HydroShare, with the ID [ebda23b9e6054f5d8e6dc1ddb1594d26](#)



Appendix I. Source Codes Locations

The source code involved in the entire project is freely available to the public.

There were two almost independent platforms used in the research – the app, and the Django based HydroDS web service. Hence, there are two repository that contains all the code involved in the project, which are:

For the app: <https://github.com/prasanna310/hydrotop-app>

For the HydroDS: <https://github.com/prasanna310/hydrods-dev>