A SERVICE-DRIVEN APPROACH TO ASSIST WATER MANAGEMENT DURING EXTREME EVENTS

BY

TINGTING ZHAO

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Civil Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2017

Urbana, Illinois

Doctoral Committee:

Professor Barbara Minsker, Chair
Professor Ximing Cai
Professor R. Srikant
Professor Halit Uster, Southern Methodist University

ABSTRACT

Water shortages and flooding have caused large property losses and endangered human lives in many areas. Rapid and informed response is needed to ensure effective water management, including reliable and immediate data synthesis, near-real-time forecasting, and model-based decision support for water operations. A structure to rapidly process heterogeneous information and models needed for near-real-time water management is critical for decision makers. This dissertation develops a service-driven approach to decision support in water management, focusing on case studies related to drought and flooding.

For flood management, real-time reservoir management is a critical component of decision support. Estimating and predicting reservoir inflows is particularly essential for water managers, given that flood conditions change rapidly. We propose a data-driven framework for real-time reservoir inflow prediction, using a service-oriented approach, that enables ease of access through a Web browser. We have tested the services using a case study of the Texas flooding events in the Lower Colorado River Basin in November 2014 and May 2015, which involved a sudden switch from drought to flooding. We have constructed two prediction models: a statistical model for flow prediction and a hybrid statistical and physics-based model that estimates errors in the flow predictions from a physics-based model. The performances of these two models are compared for short-term prediction. In addition, both the statistical and hybrid models have been published as Web services through Microsoft's Azure Machine Learning (AzureML) service, and are accessible through a browser-based Web application. The study demonstrates that the statistical flow prediction model can be automated and provides acceptably accurate short-term forecasts. However, for longer-term prediction (2 hours or more), the hybrid model fits the observations more closely than the purely statistical or physics-based prediction models alone.

The second case study focuses on droughts, developing methods to better manage significant imbalances between water supply and demand. A service-driven approach is used to couple river modeling services with optimization services for determining optimal water allocation strategies under daily drought scenarios in a permit system. An accurate and computationally efficient meta-model approach is then developed to relieve the computational burden of the simulation-optimization model. This work uses a drought event in the Upper Guadalupe River Basin, Texas, in April 2015 as a case study to illustrate the benefits of the

approach. Weather and water demand uncertainty are considered through scenario-based optimization. The results have demonstrated that the simulation-optimization model services can easily be coupled using DataWolf workflow tool and AzureML service, providing improved water allocation strategies relative to the current approach. The scenario analysis shows that the permit grouping system, which organizes water right permit holders into groups rather than considers each water user individually, is an easy and manageable approach for water allocation. In addition, the adaptive meta-model approach is efficient to relieve the computational burden in simulation-optimization model, thereby enabling large-scale real-time Web services for decision support.

ACKNOWLEDGEMENTS

I would like to express my great appreciation to my advisor, Professor Barbara Minsker, for her guidance in finishing my graduate study. I greatly appreciate all the support she gave me to make it through the frustrating moments during PhD study, and for helping me think critically about my research, and endowing me with freedom to develop my interests. Thanks for being my mentor by listening, talking and acting as an inspiration in my youth.

I would also like to thank my co-advisor, Professor Ximing Cai and other members of my thesis committee, Professor R. Srikant and Professor Halit Uster, for providing me with suggestions during the PhD study.

I would like to thank my managers and mentors during my internship in Microsoft for their encouragement and inspiration in exploring my future career in data science.

I would like to credit Jong Lee and Christopher Navarro from NCSA who greatly helped me in setting up model services in the study. I especially appreciate Chris's support and help in developing and implementing RAPID model services.

I would also like to express my gratitude to the folks from Lower Colorado River Authority (LCRA) and the Texas Commission on Environmental Quality (TCEQ) for providing research data.

I am truly grateful for the ten years' friendship with Rina Sa and Taoran Song. No matter where we are, we always support each other. I also wish to express my sincere gratitude to my good friends, Xing Gong, Yiming Li, Daisy Chen, Wendy Xu and others, for all the support and fun times.

I am indebted to my group members in Professor Barbara Minsker's group and Professor Ximing Cai's group. All the countless hours of help are greatly appreciated.

Furthermore, I acknowledge a gift from Microsoft Research that provided funding support.

Finally, I would like to express my deepest thanks to my parents for their encouragement and love.

CONTENTS

Chapter 1

Introduction and Thesis Summary

Water shortages and flooding have caused large property losses and endanger human lives in many areas. Rapid and informed response is needed to assist in water management. Effective water management involves reliable and immediate data collection, model-based water operations, and other information. Take Texas' drought and flooding events as an example. Texas experienced a severe drought in 2011 and the drought lasted four years. Then the same area underwent a sudden switch from drought to flooding in November 2014 and May 2015.

When flooding happens, reservoir operations play an important role in flooding control. During management of a flood event, reservoir inflow information serves as a base for reservoir operations. Previous studies have focused on predicting reservoir inflows from rainfall and historical reservoir inflows using physics-based models. They have not incorporated soil moisture as an input feature. Toukourou et al. [2010] showed that rainfall and soil moisture data are the major relevant variables for reservoir inflow. Therefore, this work explores which type of data-driven approach can be applied to improve real-time reservoir forecasting and investigates the role of soil moisture in predicting reservoir inflows.

Droughts have highlighted the fact that water users are facing significant imbalances between water supply and demand. Additionally, water allocation strategies are constrained by the uncertain future conditions of climate and water demand. The simulation-optimization approach has been used to account for complex water allocation problems, but coupling disparate models in a framework has been an obstacle for effective water management. This work demonstrates the promise of coupling simulation-optimization model services to improve real-time water management and implementing a service-driven framework to identify the best water allocation strategies for different drought scenarios in a priority permit system.

The computational burden of the simulation-optimization model, which stems from the complex constraint evaluation, becomes a major challenge for large-scale real-time water management. An accurate and computationally efficient approach for conducting complex constraint evaluation is explored and different handling approaches are compared in terms of optimization performance and accuracy.

The service-driven approach for each model is deployed in a loose-coupling environment to view each model service as an individual component and exchange information among each component over the whole network. The modeling component that requires complex configuration and specific data standards can be integrated into a large decision support system with each model located in its own running environment and easily accessed through Web services.

This chapter summarizes the service-driven approaches developed in this thesis to predict reservoir inflows during floods (Section 1.1) and to improve drought management using simulation-optimization models (Section 1.2). Meta-Model methods for constrained non-linear optimization are also developed and tested to improve the computational efficiency of simulation-optimization methods (Section 1.3).

1.1 A service-driven approach to predict reservoir inflows during floods

Reservoir management is a critical component of flood management, and information on reservoir inflows is particularly essential for reservoir managers to make real-time decisions given that flood conditions change rapidly. In this work, we demonstrate a new framework for real-time flood management through data-driven services to rapidly estimate reservoir inflows from available data and models.

Traditional hydrologic models have evolved from lumped conceptual models to physics-based distributed models where approximations of the partial differential equation or empirical equations are applied [*Abbott et al.,* 1986b]. Models of the physical processes employ mathematical functions that simulate hydrologic processes and usually involve complex nonlinear processes with high spatial variability at the basin scale [*Islam*, 2011]. Physics-based models are widely used in reservoir management. For example, the National Weather Service (NWS) river forecast centers use physics-based models for daily forecasts. These models often require extensive manual effort for calibration that can make real-time updates difficult. Data sources for physically-based models can be complex and limited, and calibration can be difficult and time-consuming. Data-driven modelling is an alternative approach that allows rapid construction of complex models to estimate outcomes based on past experiences and historical events.

Data-driven models, such as statistical or machine learning models, use historical data to rapidly learn a functional map between concurrent input and output variables. Large and growing volumes and varieties of data can be retrieved to derive these types of models using data services from sensors, satellites, and other data sources. Data-driven models can be coupled with physics-based models by fitting a data-driven model to the residual error from the physics-based model, thereby reducing any persistent bias in the physics-based model [*Singh & Woolhiser,* 2002].

Previous studies have focused on non-linear regression models, which have good predictive performance in comparison with other statistical models. Although previous studies have focused on predicting reservoir inflow from rainfall and historical reservoir inflow data, they have not incorporated soil moisture as an input feature. Toukourou et al. [2010] showed that rainfall and soil moisture data are the major relevant variables for reservoir inflow.

In this work, we propose a data-driven framework for real-time reservoir inflow prediction using a service-oriented approach that enables ease of access through a Web browser. Statistical and hybrid models are developed to predict flow and residual errors from a physics-based model, respectively. We use boosted regression trees (BRT) as function approximators to predict reservoir inflows from real-time and historical precipitation and soil moisture data. Our early tests showed that, for this application, BRT has the advantages of faster training and higher accuracy than ANN. Some literatures have recently shown that BRT is effective as an ensemble machine learning approach for hydrology. The data-driven models are developed using Azure Machine Learning (AzureML) Studio, a Cloud-hosted predictive analytics software toolkit that allows for the graphical construction of data pipelines in a user-friendly Web browser interface for data requests, fitting predictive analytics models, and data visualization [*AzureML team Microsoft,* 2015]. The models built in AzureML Studio are then published as Web services in the Azure Cloud, providing scalability and high software availability and reliability, as well as easy integration into modern software systems.

The results of this work demonstrate that the statistical flow prediction model can be automated and provides acceptably accurate short-term forecasts. However, for longer-term prediction (two hours or more), the hybrid model fits the observations more closely than the purely statistical or physics-based prediction models alone. Both the flow and hybrid prediction models have been published as Web services through Microsoft's Azure Machine Learning

(AzureML) service and are accessible through a browser-based Web application, enabling ease of use by both technical and non-technical personnel.

1.2 A service-driven approach to manage water allocation during droughts

In recent years, water shortages have been frequent occurrences in the United States. Texas experienced its most serious drought during the summer of 2011. The devastating drought caused a $5.2 million loss in Texas agriculture [*Susan Combos,* 2012] and endangered human life. California is also facing one of its most severe droughts since 2013. The decline of river basin aquifers and groundwater levels, along with population increases, has raised water allocation issues. The imbalance between water supply and water demand poses a crucial question for water management. Effective water management requires inputs from multiple climate, river, and optimization models. The inability of these models to communicate with one another is an obstacle for operations because the model languages and input-output data formats are different for each model. This study proposes a service-driven approach to couple river modeling services with optimization services to provide water managers with effective decision support processes.

Since the simulation-optimization model is a complex nonlinear formulation that cannot be solved by traditional mathematical optimization methods, metaheuristic approaches are popular tools in water management to help decision makers with new water management strategies [*Maier et al.,* 2014]. Genetic Algorithm (GA), a type of metaheuristic evolutionary algorithm, is implemented to solve simulation-optimization problems in water management.

This study's purpose is to develop a coupled simulation-optimization service for water allocation during droughts. The simulation-optimization model has been used widely in water resource management, but this would be the first service-driven approach for simulation-optimization models in which the optimization service can communicate with the simulation service. The built Web services are published as a Web application that enables near real-time water decision support. The study uses the drought event in the upper Guadalupe river basin in April 2015 as a case study. The Texas Commission on Environmental Quality (TCEQ) is responsible for water allocation management in the river basin. Currently, they allocate water based on subjective judgment without a rigorous science-based approach. Coupled simulation-optimization services can assist TCEQ water managers with an effective decision-making

process based on the best available forecasts of river streamflow.

This work presents a service-driven framework using coupled simulation-optimization models and demonstrates its application for identifying optimal water allocation strategies under each drought scenario in a permit system. Scenario analysis results show how the curtailment hours for each group of water right holders in the TCEQ permit system respond to the uncertainty of climate and water demand. The uncertainty has different impacts on each group of water right holders and the results can assist decision makers in adopting more effective water allocation management strategies. An alternative TCEQ Priority Doctrine that extends the water users to more groups is proposed and proved to be beneficial to junior water users. In addition, non-compliance in optimal water allocation by junior water right holders has a greater effect on the river system than non-compliance by senior water right holders. In addition, robust scenario analysis is explored and the results suggest that the most senior water right holders can make full use of water, more junior groups are more susceptible to varying levels of water usage depending on current conditions, and the most junior groups are completely cut off from water use under all scenarios. Overall, the application of the built simulation-optimization service provides a simple and practical approach for water managers to obtain a more robust water allocation strategy under a number of different scenarios.

1.3 Meta-Model methods for constrained complex optimization during droughts

More frequent droughts resulting from climate change are increasing imbalances between water supply and water demand [*Wilhite & Glantz,* 1985]. A Web service-driven framework for water management during droughts has been built to couple simulation and optimization services. However, the computational effort in handling the constraints, which involves running computationally-intensive models repetitively, is a major obstacle to developing an effective real-time Web application for decision support.

The objective of this work is to relieve the computational burden in constrained non-linear optimization problems by applying meta-model approaches, thus enabling large-scale real-time Web services for decision support during droughts. If the constraint in non-linear optimization problems consists of complex mathematical equations or contains parameters

calculated from complex simulation models, the computational burden of evaluating the constraint becomes a major challenge.

This work compares offline meta-model and online meta-model training approaches using different machine learning classifiers to rapidly judge whether a constraint is satisfied. Conservative meta-models that weigh feasibility more than accuracy are developed to guide optimization exploration in the feasible region. Different classifiers (support vector machine, neural network, logistic regression, and an ensemble of these classifiers) are tested in this work. The performance of the online meta-model approach does not depend on the choice of classifier. The results have shown that the offline training approach converges to a near-optimal solution while surpassing the online training approach in computational efficiency. The best-performing online meta-model, whose performance is independent of the choice of machine learning model, converges to the optimal solution and saves approximately 60% of the computation time.

Finally, the meta-model service is built and coupled with the new optimization service for a real-time Web application. Previous meta-model approaches mainly focused on replacing the simulation models without considering the model's role in the optimization process. The proposed meta-models, which are specifically designed to efficiently evaluate the constraint of an optimization model, are a novel development of this project that will support real-time large-scale Web applications of non-linear optimization services. The real-time Web application is the first to assist water managers with real-time information retrieval and a scientifically-valid modeling approach to improve water allocation strategies.

Chapter 2

Literature Review

This chapter provides a review of the background literature for the research presented in subsequent chapters. It covers the data-driven model application in water management, the simulation-based optimization model, the meta-model-based Genetic Algorithm, and the service-driven approach in model application.

2.1 Service-driven Approach

Decision support in water management involves different types of models, such as data-driven prediction models, simulation models, optimization models, etc. Existing disciplinary models can be written with different configurations and in different programming languages that may have difficulty in communicating. A structure for organizing heterogeneous information and applying it to different models to rapidly solve water management problems is critical for decision makers [*Laniak et al.,* 2013]. Model as a service (MaaS) has recently emerged as an efficient tool to solve the above challenges.

Model as a Service (MaaS) originates as a merging of Software as a Service (SaaS) and the Model Web [*Roman et al.*, 2009]. SaaS involves creating services that deliver a software application through Web browsers, allowing users to easily access the software, share data, and improve interoperability [*Roman et al.*, 2009]. Each model is still run in its own configuration environment and the functionality of each model is published on the Web [*Goodall et al.,* 2011]. Model Web is an open-ended system for interoperating models and data with access to machines and the Internet through Web services [*Geller & Turner*, 2007]. MaaS merges the two approaches to create an automated modeling system for data access, model execution, and output visulization through the Web, using standard data formats for data interoperability [*Roman et al.*, 2009]. The model execution does not require specific skills and the output can be viewed directly on the Web through visualizaton tools.

The service-driven approach has been implemented for decision support in water management. Goodall et al. [2008] proposed a Web services approach in the water resources management area for the National Water Information System, using Web services to easily access hydrologic data through a standard protocol as well as to interoperate among disparate

data sources. Fang et al. [2013] presented a water information system prototype that includes geoinformatics, enterprise information systems, and cloud services and integrates data management, simulation modeling, and knowledge management into the information warehouse. Such an approach is significant to effective water resource management. Almoradie et al. [2013] adopted Open Geospatial Consortium (OGC) WaterML 2.0 standard to deliver water-related data through the Web, resulting in a Web-based flooding information system. Sun [2013] proposed service-oriented computing to reduce the computation burden of decision support systems in watershed management. Ames et al. [2012] presented Web services-based software incorporating hydrological data access, data visualization, and data analysis. Jones et al. [2015] developed a cloud-based MODFLOW groundwater modeling service for aquifer management decision support. The MODFLOW simulation model service is an automated system that imports user input, executes the model, and visualizes results as maps for groundwater management. These studies have each focused on a single data, model or computing service; however, these model services have not yet been coupled. Therefore, this work extends the service-driven approach to the coupling of disparate model services.

2.2 Data-driven Model in Water Management

Advances in information technologies allow for the automation of data acquisition, analysis, and visualization. Data-driven modelling, especially machine learning, allows for the construction of complex models for estimating outcomes based on experience and historical events. Rather than deriving mathematical equations from physical processes, data-driven models analyze concurrent input and output time series [*Solomatine and Ostfeld*, 2008]. The popular data-driven methods used in river systems include artificial neural networks (ANN), fuzzy rule-based systems, and support vector machines (SVM), among others.

Many applications of ANN focus on rainfall-runoff models [e.g., *Abrahart et al*., 2007, *de Vos and Rientjes*, 2007, *Nourani et al*., 2009]. Rainfall is a common input feature for data-driven models of river systems. Many reservoir inflow prediction studies also rely mainly on ANN and rainfall data. Coulibaly et al. [2000] first used an ANN to forecast daily reservoir inflow and a multi-layer feed-forward neural network (FNN) with an early stopped training approach (STA) to improve prediction accuracy. EI-Shafie et al. [2007] used historical reservoir inflow and ANN to predict monthly reservoir inflow. Bae et al. [2007] implemented an Adaptive

Neuro-Fuzzy Inference System (ANFIS) to predict monthly dam inflow using previously observed data and future weather forecasting information. Jothiprakash and Magar [2012] predicted daily and hourly intermittent rainfall and reservoir inflow using ANN, ANFIS, and linear genetic programming (GLP). Valipour [2013] compared autoregressive moving average (ARMA) and autoregressive integrated moving average (ARIMA) using increasing numbers of parameters with static and dynamic artificial neural networks. With historical time series data as input, they demonstrated that static and dynamic autoregressive ANNs perform best in forecasting monthly reservoir inflow. Kumar et al. [2015] developed an ensemble model based on neural networks, wavelet analysis, and bootstrap data sampling to generate a range of forecasts instead of point predictions for reservoir inflow.

As described above, ANN is a common hydrological approach (see also *Bowden et al*., 2012, *Abrahart et al.*, 2012, *Maier et al*., 2010), but its convergence speed is low, and training can require significant time, which may prove a barrier when near-real-time model updating is required [e.g., *Jain et al*., 1999, *Maier & Dandy*, 2000]. Others have recently shown that BRT is effective as an ensemble machine learning approach for hydrology. Erdal and Karakurt [2013] have applied BRT as an ensemble learning method that performed well in predicting a monthly streamflow forecast. Snelder et al. [2009] have used BRT to map the flow regime class by predicting the likelihood of the class of gauge stations based on watershed characteristics. BRT has the advantages of regression trees (which are based on decision trees and built on a process of recursive partition) and boosting methods (which create ensembles of multiple models that combine fast but weak learners to create a strong learner). The approach combines multiple simple trees into an additive regression model to improve predictive performance [Elith et al. 2008]. Therefore, this study uses BRT as function approximators.

2.3 Simulation-based Optimization Model

Simulation–optimization models have been used widely to solve real-world water management problems. Optimization models demonstrate good results when used with simulation models [*Singh*, 2014]. The simulation models are used to simulate different scenarios, while the optimization model is essential for identifying the optimal solution under different scenarios [*Singh*, 2014]. Gaur et al. [2011] developed a simulation-optimization model for groundwater management problems. The particle swarm optimization (PSO) algorithm was used

for a multi-objective optimization model. Nazari et al. [2014] presented a simulation-optimization approach to optimize the water supply for an urban water system under an extreme scenario. They used a multi-objective genetic algorithm as the optimization module. Ebrahim et al. [2015] used successive linear programming and the nondominated sorting genetic algorithm (NSGA-II) with a simulation model to maximize the recharge and extraction rates of managed aquifer recharge (MAR) in a catchment. The study shows that the approach is efficient in evaluating MAR in a coastal aquifer. Rasekh and Brumbelow [2015] studied a dynamic simulation–optimization model to manage contamination problems in urban water distribution systems. The study couples a dynamic evolutionary optimization approach with a simulation model.

Genetic algorithms (GA), a type of population-based metaheuristic evolutionary algorithm, has been commonly applied in water planning and management [*Maier et al.*, 2014, *Nicklow et al*, 2009]. GA, developed by Holland [1975], originates from mimicking natural selection through generations of stochastic searching for a better solution. Starting from a population of individual solutions, GA conducts a guided search for an optimal solution [*Goldberg & Holland*, 1988].

Compared with other traditional optimization methods, GA can be used to solve any type of problem without mathematical derivation. Kaini et al. [2012] coupled a GA with a soil and water assessment tool (SWAT) to optimize construction costs while satisfying water quality treatment levels at a watershed scale. Tabari and Soltani [2012] developed a multi-objective model to maximize system reliability and minimize water supply costs in a water distribution system; their study compares the non-dominated sorting genetic algorithm (NSGA-II) and sequential genetic algorithms (SGAs). Chang et al. [2010] adopted a constrained genetic algorithm (CGA) to optimize reservoir storage while considering ecological base flow as constraints in multi-use reservoir operation management. Andrea et al. [2010] implemented a GA algorithm to minimize combined sewer overflows (CSOs) for real-time decision support. The memory approach in the GA algorithm was applied to speed convergence to the optimal solution. These previous studies demonstrated that GA is a powerful tool for solving complex water management problems [*Nicklow et al.*, 2010]. Therefore, the GA algorithm is applied in this work to solve the simulation-optimization problems.

The studies described above have executed simulation-optimization models offline and have not yet explored a service-driven approach. In this work, the coupling of the optimization service with other simulation services is developed. A user-friendly Web application based on the model services allows TCEQ decision makers to allocate water through a more scientific approach.

2.4 Meta-Model-based Genetic Algorithm

The simulation models simulate the physically-based process using mathematical equations and the intensive computational budget due to the detailed representation of the real-world systems. Multiple runs of the simulation models in real-world application have become an obstacle in water resource simulation-optimization problems [*Razavi et al.*, 2012a]. The meta-model approach, which replaces the simulation model with an approximating surrogate function, has been applied in simulation-optimization models to improve computational efficiency [*Yan & Minsker*, 2011, *Pasha & Lansey*, 2010, *Gu et al.*, 2011]. The meta-model is built using the input-output dataset of the simulation models. There are two methods for the meta-model approach that are commonly used in the water resources field [*Razavi et al*., 2012b]: 1) a meta-model is built using a large training dataset before the start of the optimization [*Johnson & Rogers*, 2000, *Cai et al.,* 2015]; 2) an adaptive meta-model is initialized using a small training dataset and is iteratively updated during the optimization process [*Yan & Minsker*, 2011, *Wu et al*., 2015]. Cai et al. [2015] implemented the support vector machine (SVM) as the statistical surrogate model to replace the complex simulation model SWAT (Soil and Water Assessment Tool). They built a surrogate model under each climate scenario using the training dataset from SWAT. Yan & Minsker [2011] applied a dynamic meta-model to replace Monte Carlo simulations within a noisy genetic algorithm in groundwater remediation problems. Wu et al. [2015] developed an adaptive surrogate model to replace integrated surface water (SW) and groundwater (GW) models in water management optimization. The computational cost efficiency of the surrogate models assisted in finding the solution to comprehensive basin-scale optimization and water management problems, which fills the gap between complex environmental models and real-world water management.

Previous meta-model approaches mainly focus on replacing the simulation models without consideration of the model's role in the optimization process. This work is the first to

construct a classification model to evaluate constraint in meta-model application. It compares the performance of offline training and online training approaches. Viana and Haftka [2012] and Razavi et al. [2012b] have suggested the ensemble model to average the performance of different regression models in previous meta-model approaches. The ensemble classification model using majority voting in meta-models is proposed. The conservative meta-model in offline training approach, which increases the threshold of probability to determine the class label, is firstly applied to assist in exploring feasible solutions.

The service-driven approach described in section 2.1 has been applied in each of the model implementations above. Chapter 3 develops data-driven model services that integrate data services with the data-driven model service. Chapter 4 focuses on the coupling of simulation model services and optimization model services. The meta-model service in Chapter 5 is developed to replace the simulation model service in Chapter 4 to support real-time large-scale water management.

Chapter 3

A Service-driven Approach to Predicting Reservoir Inflows During Flood Events

Chapter 1 and Chapter 2 has introduced the background of the data-driven model services in flooding management. This chapter presents the data-driven framework for reservoir inflow prediction using a service-driven approach (Section 3.2), which includes data preprocessing, model construction, and Web application. Implementation of the framework is demonstrated with a case study in the Lower Colorado River Basin (Section 3.3). Results and discussions are presented in Sections 3.4 and 3.5.

3.1 Introduction

This study's purpose is to investigate the feasibility and accuracy of real-time data-driven services to estimate reservoir inflows from available data. The Texas flooding events in the Lower Colorado River Basin in November 2014 and May 2015, which involved a sudden switch from drought to flooding, are used as a case study. The Lower Colorado River Authority (LCRA), which is responsible for reservoir management in this basin, uses the physics-based Hydrologic Engineering Center's Hydrologic Modeling System (HEC-HMS) in the Corps Water Management System (CWMS). HEC-HMS predicts reservoir inflows from real-time data, including precipitation, reservoir information, and other hydro-meteorological data.

Currently LCRA uses a HEC-HMS rainfall-runoff model to predict reservoir inflows that does not consider soil moisture as an input dataset. The observed streamflow and soil moisture data are used only to calibrate reservoir inflows manually. Soil moisture may be an important factor for predicting reservoir inflows [*Kang et al.,* 2015] and a data-driven approach would allow LCRA reservoir managers to automatically update the reservoir inflows as these conditions change. In this study, we explore a workflow approach that allows the model set-up process to be completed only once by a technical analyst and then executed by technical or non-technical users through a Web browser. A workflow is a collection of tasks that build an automated pathway for heterogeneous modeling steps.

The performance of data-driven modeling approaches, including both statistical and hybrid (coupling statistical and physics-based) models is also assessed using boosted regression tree modules from AzureML to predict reservoir inflows from real-time and historical

precipitation and soil moisture data. The models can be easily connected with other data services to obtain the input data. The system is implemented as Web services on AzureML, which do not require any software installation and can be rapidly updated as new data are obtained. The data-driven services allow users and water managers to automatically fit model parameters, compute data-driven models, and retrieve reservoir inflow information through a Web browser.

3.2 Methodology

Figure 3.1 shows the general data-driven framework developed in this study to support reservoir management. The framework consists of two main components: 1) algorithms and tools from Azure Predictive Analytics toolkit; and 2) Web application. Azure Predictive Analytics (predictive analytics is a commercial term for machine learning) is a machine learning platform that allows rapid training of statistical models to describe the relationships between inputs ("features") and outputs ("targets"), with execution on remote servers (in the "Cloud"). This first component comprises data preparation, data preprocessing, and model development. The input datasets, which include feature datasets and target values, are first uploaded into AzureML Studio.

For this study, a wavelet analysis filter method is applied for data preprocessing to reduce data noise, since noise or errors in the measured datasets may mask important features in the data. Boosted Regression Tree modules in AzureML are then employed to statistically model the reservoir inflows using data-driven models. These model execution steps have been constructed as workflows in AzureML, and flow prediction models and hybrid prediction models have been implemented as modules in a workflow to predict reservoir inflow. AzureML has significant advantages in publishing the constructed workflows as Web services. A Web application, which is Web browser-based software for executing the built models, has been built that enables users to easily execute the data-driven model using Web services to predict reservoir inflow (named flowin in this study).

Data-driven models use historical data to learn a functional map between input and output variables that can be used to predict future output variables. Given input datasets that include input features and output target values from historical data, a mapping can be built to predict future outputs from known future input features [*Mitchell*, 1997]. For instance, $y=f(x)$ is a mapping (training model) between input variables x and output variable y. Once the future input

variables $\hat{x}$ are available, the future outputs $\hat{y}$ can be predicted using the training model. In this study, we develop two types of data-driven models. The first type is a purely data-driven statistical prediction model that is used to directly predict reservoir inflows from soil moisture, precipitation, upstream reservoir outflow, and historical reservoir inflow. The second type of model is a hybrid prediction model, which corrects the results of physics-based models that predict reservoir inflows from weather, runoff, and streamflow predictions. The hybrid prediction model applies the available input features to predict differences between the physics-based model-predicted results and the observed data.

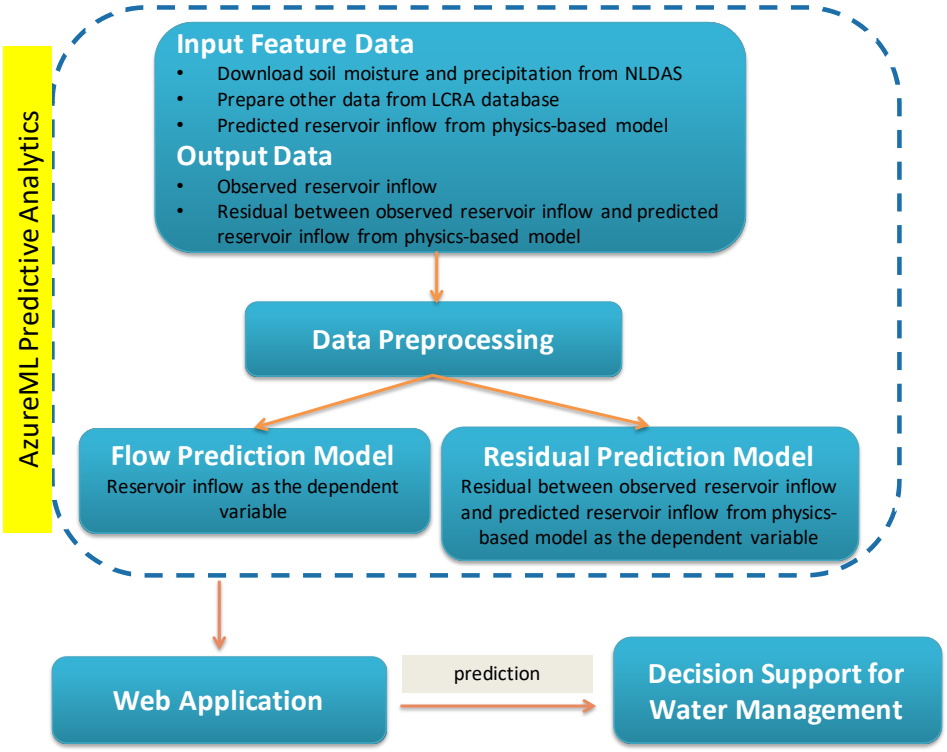

Figure 3.1. Framework of the Predictive Analytic Services

3.2.1 Data Preprocessing Using Wavelet Analysis

Wavelet analysis filters the reservoir inflow data into trend and noise parts, a necessary step because either wave action from high storm winds or sensor measurement errors can cause fluctuations that affect measured reservoir inflow data. We use wavelet functions to decompose the original data into high-pass filter (details) and low-pass filter (trend) components [*Valens*, 1999, *Polikar*, 2001, *Okkan*, 2012].

Maximal Overlap Discrete Wavelet Transform (MODWT) is a linear filtering operation that produces time-dependent wavelets and scaling coefficients [*Cornish & Percival*, 2005]. It performs better than other methods, such as discrete wavelet transform (DWT), in fitting all sample sizes because DWT requires sample size to be a multiple of 2^J where J is the decomposition level [*Cornish & Percival,* 2005]. In addition, MODWT is independent of the starting point of the time series, which means that MODWT is not affected by circular shifting of the input time series [*Percival and Walden*, 2000].

The high-pass-filter-generated wavelet coefficient is defined as

$$\overline{W}_{j,t} = \sum_{l=0}^{L_j-1} \tilde{h}_{j,l} X_{t-l} \tag{3.1}$$

and the low-pass-filter-generated wavelet coefficient is defined as

$$\bar{V}_{j,t} = \sum_{l=0}^{L_j-1} \tilde{g}_{j,l} X_{t-l} \tag{3.2}$$

where j is the level of decomposition, L is the width of the j=1 base filter, and $\{\tilde{h}_{j,l}\}$ and $\{\tilde{g}_{j,l}\}$ are wavelet and scaling filters respectively.

The decomposition process is shown in Figure 3.2. Take the decomposition level = 3 as an example. In each level, the original dataset X is decomposed as trend V and residual error W. In the first level, X is decomposed as V1 and W1. The level 2 decomposition is based on V1, which is the trend component from the last level. W1 is discarded. The decomposition continues until the defined decomposition level is reached. The level of filtering selected for the particular case study (in this case, level 2) is then selected based on best professional judgment of the reservoir operators.

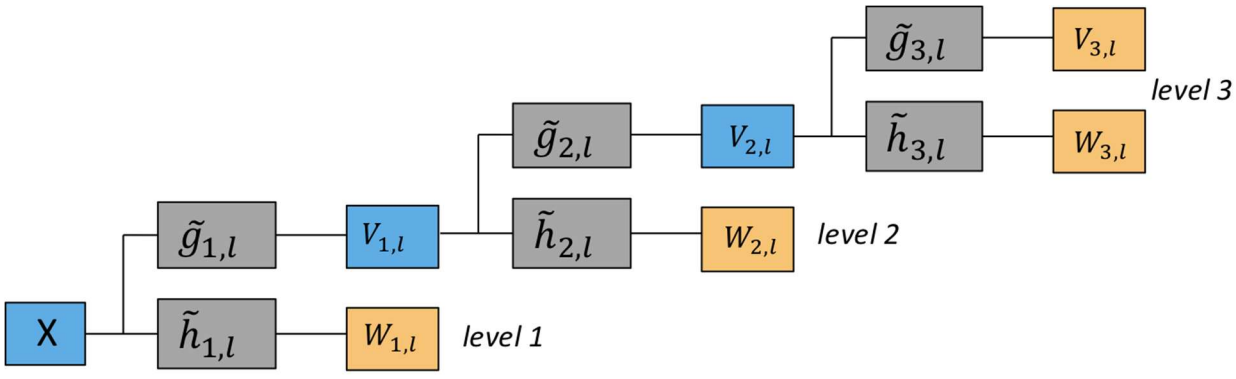

Figure 3.2. Decomposition based on Wavelet Analysis

3.2.2 Prediction Modeling using Boosted Regression Tree (BRT)

Data-driven prediction models are computed using a boosted regression tree model, which is an ensemble model that integrates multiple single regression trees. Regression tree models use recursive binary splits to predict the target variable [*Elith et al*., 2008]. Figure 3.3 demonstrates a simple regression tree example. A tree model is built by splitting the input datasets into subsets based on each selected input feature (such as x_1, x_2, x_3, x_4, x_5). The best partition (e.g., $x_1 < V_1$ and $x_1 \geq V_1$) is computed from each derived subset (called recursive partitioning) to maximize improvement in the model prediction. This process continues until no further splitting improves the predictions. Boosting is an adaptive method of combining simple models into a single strong learner to improve model performance. Pseudo code for BRT has been included in the appendix of the dissertation. Key features are the ability to fit complex nonlinear models and high accuracy [*Elith et al.,* 2008, *Caruana & Niculescu-Mizil*, 2006].

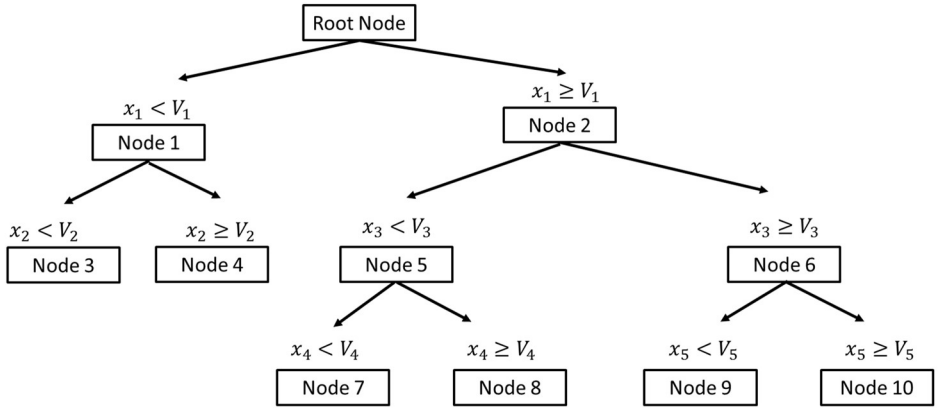

Figure 3.3. Example of a Regression Tree

3.2.3 Performance Metrics

We use five performance metrics to evaluate the developed models for predicting current and future reservoir inflows.

a. Mean Absolute Error (MAE)

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|\hat{y}_i - y_i| \tag{3.3}$$

where $\hat{y}_i$ is the prediction and $y_i$is the true value. MAE averages all of the errors in the model. When MAE is closer to zero, the model fits better.

b. Root Mean Squared Error (RMSE)

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2} \tag{3.4}$$

where $\hat{y}_i$ is the prediction and $y_i$is the true value. RMSE is a measurement of the average of the squares of the errors. RMSE=0 means a perfect fit of the model.

c. Relative Absolute Error (RAE)

$$RAE = \frac{\sum_{i=1}^{n}|\hat{y}_i - y_i|}{\sum_{i=1}^{n}|y_i - \bar{y}|} \tag{3.5}$$

where $\hat{y}_i$ is the prediction, $y_i$is the true value and $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$. RAE measures the percentage of error over the true value. RAE = 0 if there is a perfect fit.

d. Relative Squared Error (RSE)

$$RSE = \frac{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{3.6}$$

where $\hat{y}_i$ is the prediction, $y_i$is the true value, and $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$ is the mean true value.

e. Coefficient of Determination (R^2)

$$R^2 = \left(\frac{\sum_{i=1}^{n}(\hat{y}_i - \bar{\hat{y}})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(\hat{y}_i - \bar{\hat{y}})^2 \sum_{i=1}^{n}(y_i - \bar{y})^2}}\right)^2 \tag{3.7}$$

where $\hat{y}_i$ is the prediction, $y_i$is the true value, $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$ and $\bar{\hat{y}} = \frac{1}{n}\sum_{i=1}^{n} \hat{y}_i$. R^2 measures how close the data are to the fitted regression line. An R^2 of 1 indicates a perfect fit of the regression line, and an R^2 of 0 indicates that the line does not fit the data at all.

3.2.4 Web Application

AzureML is a Cloud service for machine learning experiments. The workflows are constructed as directed acyclic graphs (DAGs) in a Web-based graphical user interface that enables module operations on datasets [*AzureML team Microsoft*, 2015]. AzureML includes machine learning libraries from open source languages such as R and Python, in addition to libraries of statistical methods and other data processing operations. In addition, Azure ML allows connections to other infrastructure such as database servers to handle large amounts of data.

Machine learning models can be manipulated as data workflows by joining modules in AzureML Studio as shown in Figure 3.4. Such data workflows, including data preprocessing, model building, and results visualization, are more natural and intuitive than scripts. Non-technical users can easily implement and update the data-driven approach without requiring machine learning skills or computing expertise [*AzureML team Microsoft*, 2015]. After the

complete workflow is built in AzureML Studio, it can be published as a Web service and shared with other users as a Web application.

A Web application builds the connection between client and server to enable Cloud-based Web services to execute through a simple Web interface. For instance, a modeling Web application can be built as an automated modeling system (workflow) that includes data access, model execution and output visualization. Such a system can be published as Web services. A custom Web User Interface (UI) is then built to allow non-technical users to access the Web services and view the output directly through the Web browser.

In AzureML, a python Application Programming Interface (API) is provided to easily access AzureML Web services. A custom UI allows users to download input data and execute the prediction models, with the results made available through the UI. Reservoir managers who are not familiar with machine learning and data-driven approaches and are interested in machine learning approaches can easily use the Web application to predict reservoir inflow and compare or incorporate results from physics-based models. The Web services provide a rapid approach for reservoir managers to understand near-term impacts of current conditions on reservoir inflow and provides a proof of concept for a real-time Cloud-based system for reservoir management.

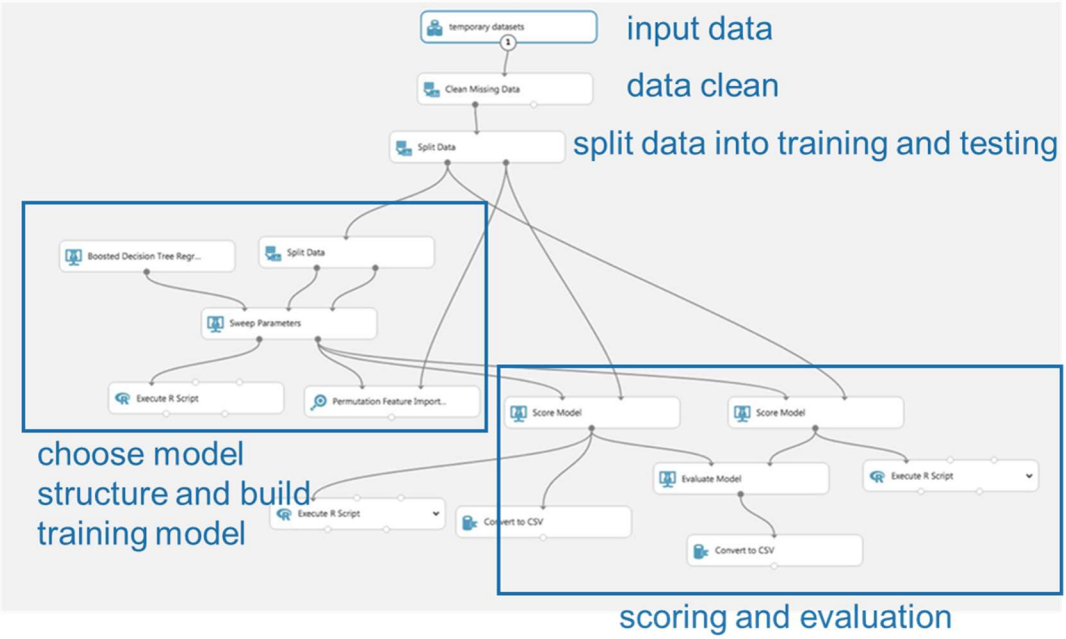

Figure 3.4. Example of AzureML Graphical Workflow

3.3 Case Study

Lake Travis is in Travis County, located in the upper stream of Lake Austin. Mansfield Dam, operated by LCRA, created Lake Travis, which serves to contain floodwaters and helps to manage flooding downstream. Lake Travis stores a maximum of 256 billion gallons of floodwaters. The floodgate release is operated by LCRA under the direction of the U.S. Army Corps of Engineers. The amount of release depends on weather and flood conditions, such as the water level of the reservoir and downstream flow. Understanding the predicted reservoir inflow during flooding events helps reservoir managers operate the dam more effectively based on such information and their operating experience [*Mateo et al.*, 2014].

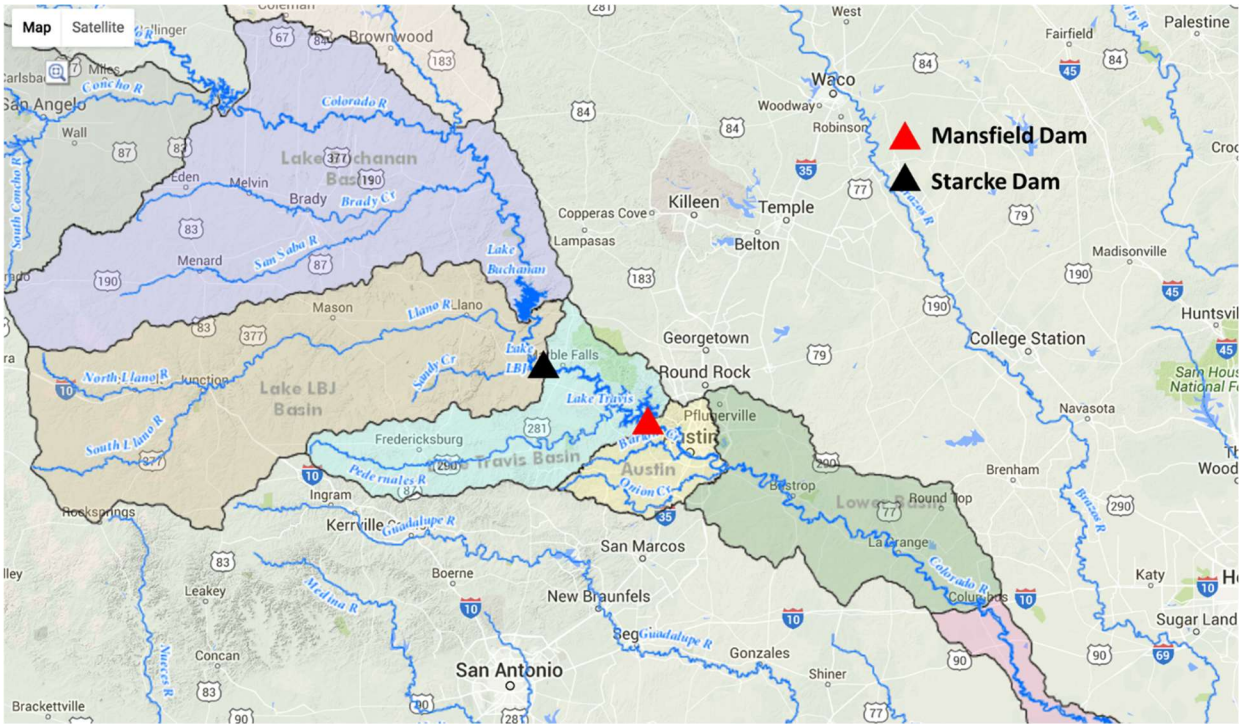

Figure 3.5.a Map of Lake Travis Basin

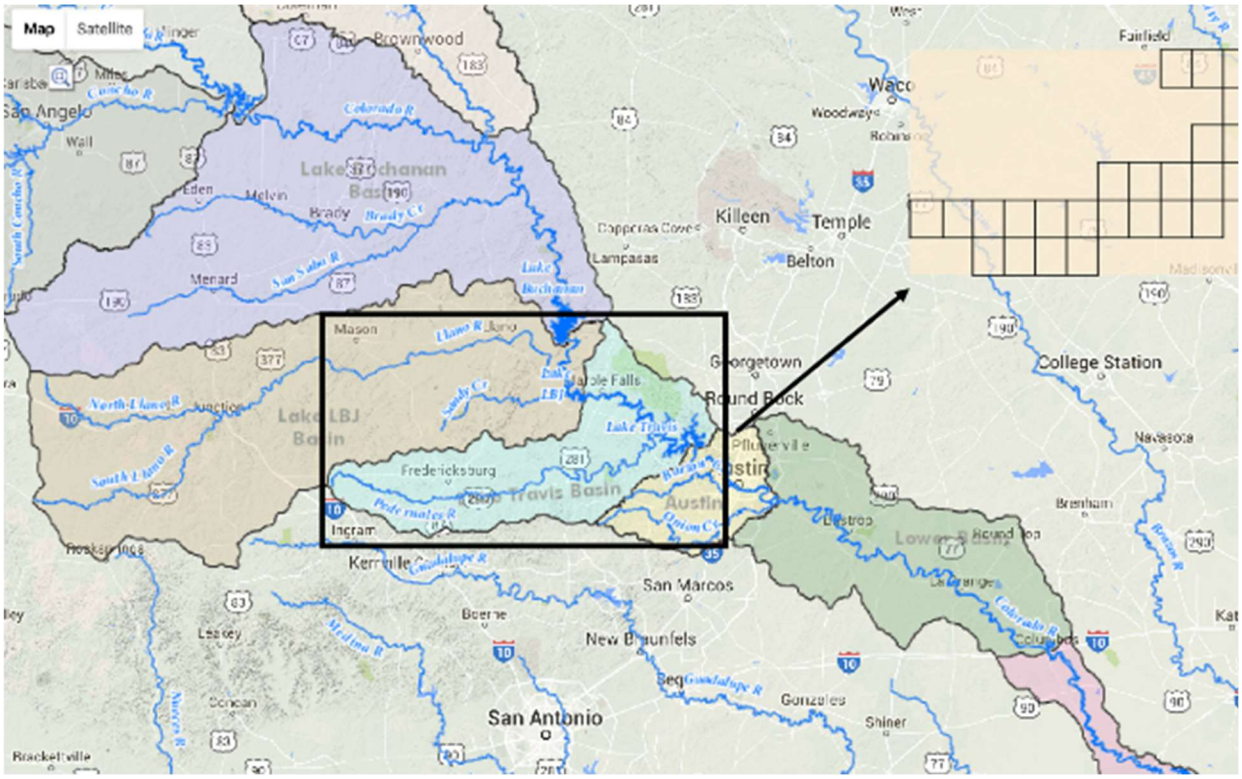

Figure 3.5.b Grid Points in Lake Travis Basin

Figure 3.5. Case Study Location and Data Points

3.3.1 Datasets

The case study focuses on Texas flooding events in the Lower Colorado River Basin in November 2014 and May 2015, using the input and output data given in Figure 3.6. Precipitation and soil moisture input data were collected from 31 grid points in Lake Travis Basin in the upper stream of Mansfield Dam, as shown in Figure 3.5.b. The precipitation becomes direct runoff and the soil moisture affects surface runoff by reducing infiltration, which physically affects reservoir inflow. Other input features are the flow out of the upstream reservoir Starcke Dam (flowout) and the previous flowin to Mansfield Dam, as shown in Figure 3.5.a.

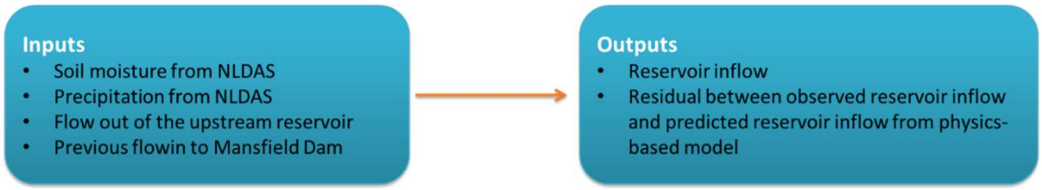

Figure 3.6. Inputs to and Outputs from the Data-driven Models

The precipitation data (in kg/m2) were downloaded from Phase 2 of the North American Land Data Assimilation System (NLDAS-2). NLDAS-2 forcing data are derived from: (1) Doppler radar data, which are used in national weather forecasts (**http://radar.weather.gov/**), (2) CPC MORPHing (CMORPH) Technique, which produces global precipitation data at a high spatial and temporal resolution **http://www.cpc.ncep.noaa.gov/products/janowiak/cmorph_description.html**), and (3) HPD (Hourly Precipitation Datasets) data (http://www.srh.noaa.gov/ridge2/RFC_Precip/). The data are in 1/8th degree grid spacing [*Rui & Mocko*, 2013]. The soil moisture data, in units of kg/m2, relied on the Noah land surface model (Noah soil moisture 0-100 cm). Data from both models can be easily downloaded via Web application by providing spatial coordinates and specific time periods.

The reservoir hourly data were collected by LCRA from November 1, 2014, 00:00, to December 3, 2014, 23:00, and from May 1, 2015, 00:00, to June 4, 2015, 23:00, which were the recent time periods with severe flooding in the Lower Colorado River Basin. These data were retrieved from the LCRA database for this study. The two flooding datasets were concatenated together. From the available datasets, the first 85% (from Nov 1, 2014 00:00 to May 26, 2015 15:00) were considered as the training dataset to train the model. The remaining 15% (from May 26th 2015 16:00 to June 4th 2015 23:00) were used for testing to evaluate the model predictions. To ensure that the validation and training datasets were interchangeable, 80% of the training dataset was designated as training and 20% as validation. The purpose of such splits is to keep the model fitting completely separate from the validation so that the model is not overfit to this particular dataset.

3.3.2 Model Implementation

Wavelet Analysis to Filter Data Noise

Wavelet analysis is intended to smooth the fluctuations in the reservoir inflow data and maintain the trend. The decomposition level (Figure 3.2) is a key element to choose in wavelet analysis. Nourani et al. [2008] estimated the optimum decomposition level for DWT using the following equation:

$$L = int[log_{10}(N)] \tag{3.8}$$

where L is the decomposition level and N is the number of time series data.

In this study, the number of time series data is 1656. Based on Equation (3.8), the decomposition level L = int[log(1656)] = 3. To select the best decomposition level, Figure 3.7 shows flowin after each level. At level 1, the dataset still has significant fluctuations and the noise removal is insufficient. At level 3, the dataset is smooth but the peak flow is significantly truncated. LCRA staff advised that Figure 3.7.b, with level 2 noise removal, represents the best data filtering: the dataset is smooth and the peak is not excessively truncated. Figure 3.8 shows the original reservoir inflow versus the filtered reservoir inflow.

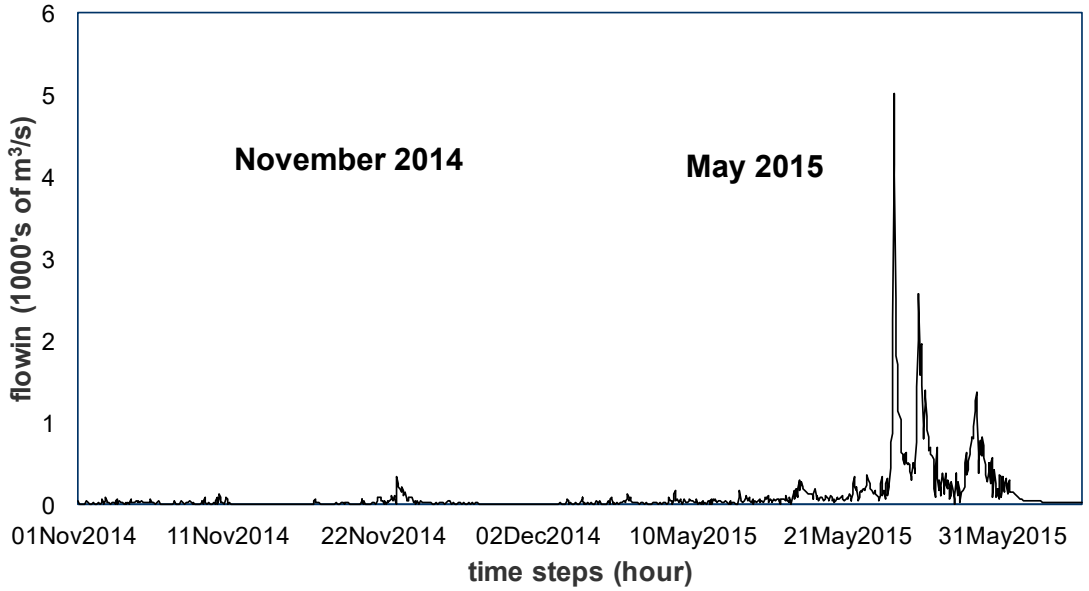

Figure 3.7.a. *Reservoir Inflow* at Level 1

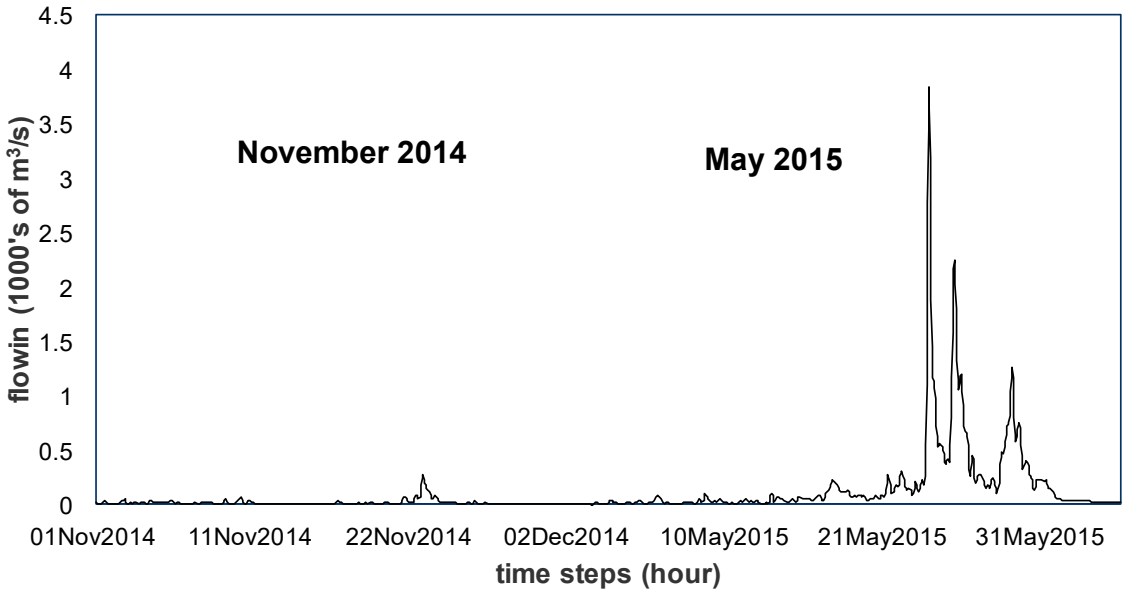

Figure 3.7.b. *Reservoir Inflow* at Level 2

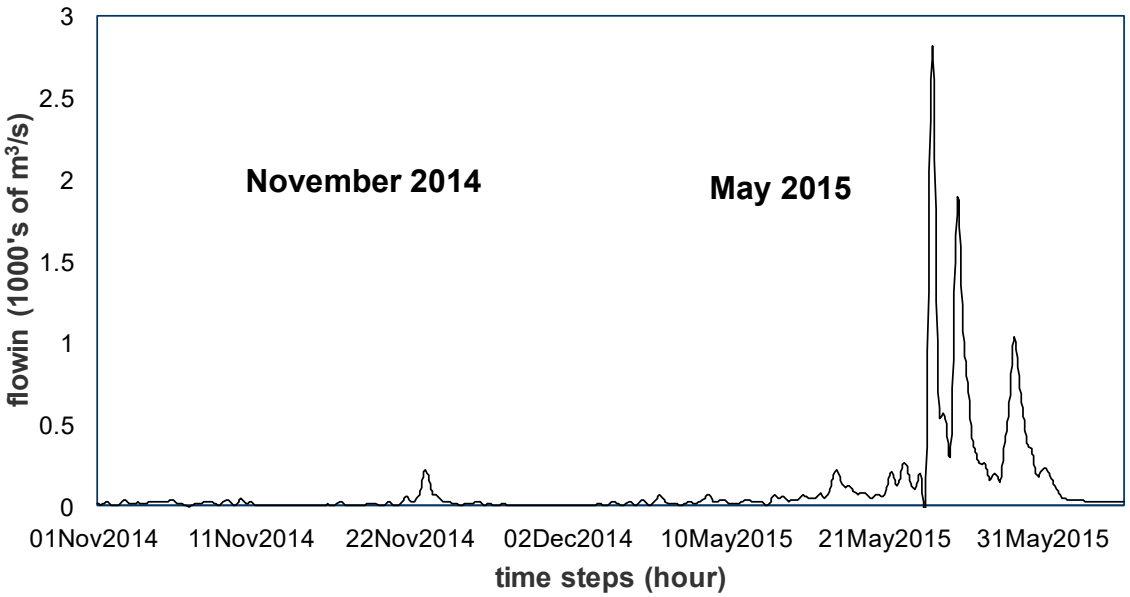

Figure 3.7.c. *Reservoir Inflow* at Level 3
Figure 3.7. *Reservoir Inflow* Graph after Each Decomposition Level

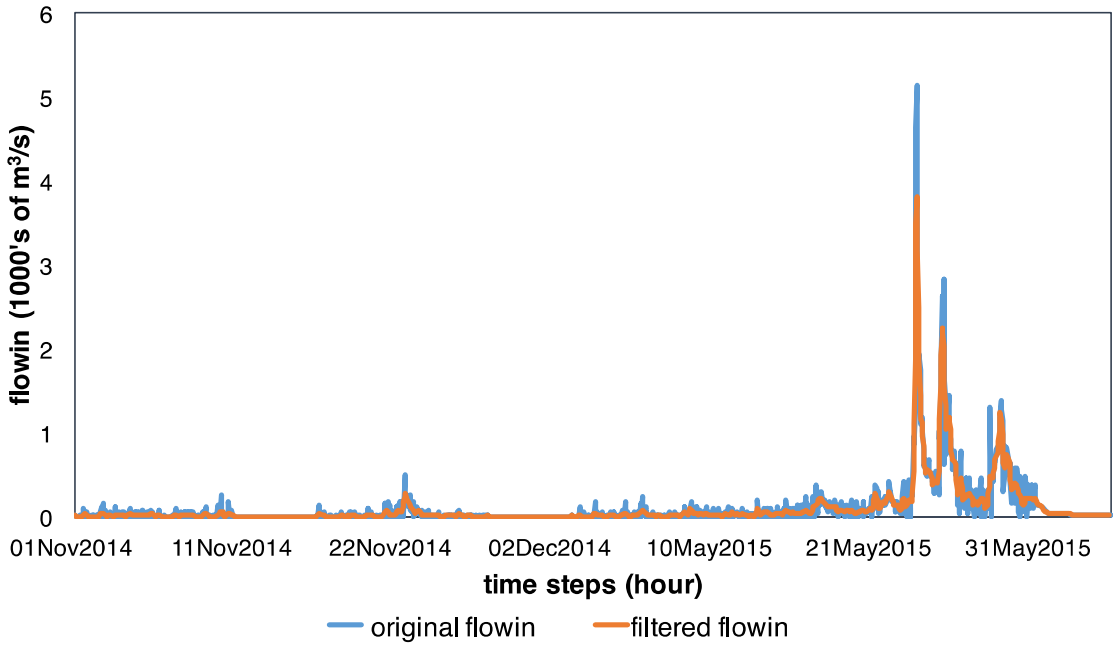

Figure 3.8. Original Observed *Flowin* vs Filtered Observed *Flowin* During 2014-2015 Flooding Events

Correlation

To assess appropriate time lags for inclusion in the model, cross correlation was performed and the results are shown in Figure 3.9. The figure presents the respective correlations between soil moisture and reservoir inflow, precipitation and reservoir inflow, and flowout from the upstream reservoir and the downstream reservoir inflow.

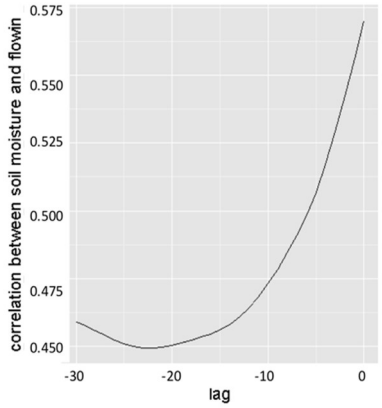

Figure 3.9.a. Correlation between Soil Moisture and *Flowin*

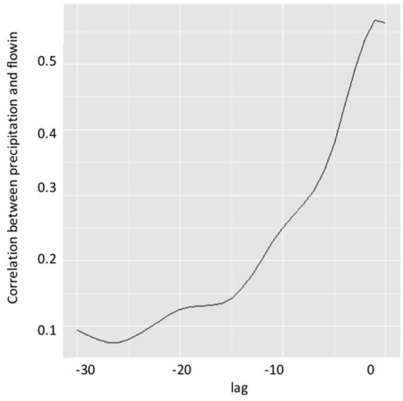

Figure 3.9.b. Correlation between Precipitation and *Flowin*

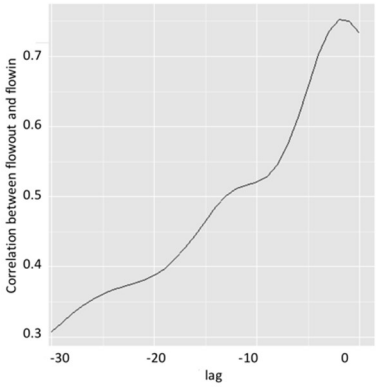

Figure 3.9.c. Correlation between *Flowout* of Upstream Reservoir and *Flowin*

Figure 3.9. Correlation Plot between Input Features and Output Label

Figure 3.9.a shows that the correlation between soil moisture and reservoir inflow reaches the highest point at lag=0, indicating that the soil moisture at time *t* is correlated most strongly with the reservoir inflow at time *t*. Figure 3.9.b demonstrates that the precipitation at time *t-1* hour affects the reservoir inflow most, as the precipitation in the past hour usually has the largest influence on the reservoir inflow. The *flowout* of the upstream reservoir (Lake Marble Falls at Starcke Dam) at time *t-2* hours is correlated most strongly with the reservoir inflow, consistent with LCRA's assessment that flow typically requires two hours to travel from the upstream reservoir to the downstream reservoir inflow at Mansfield Dam.

A flow prediction model to predict reservoir inflow

To develop the BRT model, different combinations of feature inputs were tested. Although the cross-correlation results identified the lags corresponding to the strongest correlation, experimentation with different combinations of time lags is still needed to assure the best performance. Seven experiments were conducted:

1) soil moisture at time *t* and precipitation at time *t*-1 at all 31 grid points, *flowout* from upstream reservoir at time *t*-2, and reservoir inflow at time *t*-1;

2) soil moisture at time t and precipitation at time *t*-1 at the grid point closest to the reservoir, *flowout* from upstream reservoir at time *t*-2, and reservoir inflow at time *t*-1;
3) soil moisture at time *t*-1 and precipitation at time *t*-1 at the grid point closest to the reservoir, *flowout* from upstream reservoir at time *t*-2, and reservoir inflow at time *t*-1;
4) soil moisture at time t-2 and precipitation at time t-1 at the grid point closest to the reservoir, *flowout* from upstream reservoir at time t-2, and reservoir inflow at time t-1;
5) soil moisture at time t-3 and precipitation at time t-1 at the grid point closest to the reservoir, *flowout* from upstream reservoir at time t-2, and reservoir inflow at time t-1;
6) soil moisture at time t, t-1, and t-2 and precipitation at time t-1 at the grid point closest to the reservoir, *flowout* from upstream reservoir at time t-2, and reservoir inflow at time t-1; and
7) soil moisture at time t, t-1, and t-2 and precipitation at time t, t-1, and t-2 at the grid point closest to the reservoir, *flowout* from upstream reservoir at time t-2, and reservoir inflow at time t-1 and t-2.

Since early tests indicated that the precipitation and soil moisture at the closest point to the reservoir were more predictive of the reservoir inflow, most experiments were conducted using data from the closest point to the reservoir.

AzureML facilitates ease of implementation of these alternative models using graphical workflows, shown in Figure 3.10, for data manipulation, regression models, training models, score models and other machine learning-related modules. The boosted regression tree module in AzureML was used with the following settings: maximum number of leaves per trees = 10, minimum number of samples per leaf node = 10, and learning rate = 0.1. The sweep parameter module in AzureML was used to select the number of trees constructed. Users provided a range of values for the number of trees ([5, 10, 15, 20, 30, 40, 50, 60, 70, 80] in this case) and the module builds training models for each value and selects the best (20 in this case). The criteria to choose the best number of trees was based on the MAE of the validation dataset.

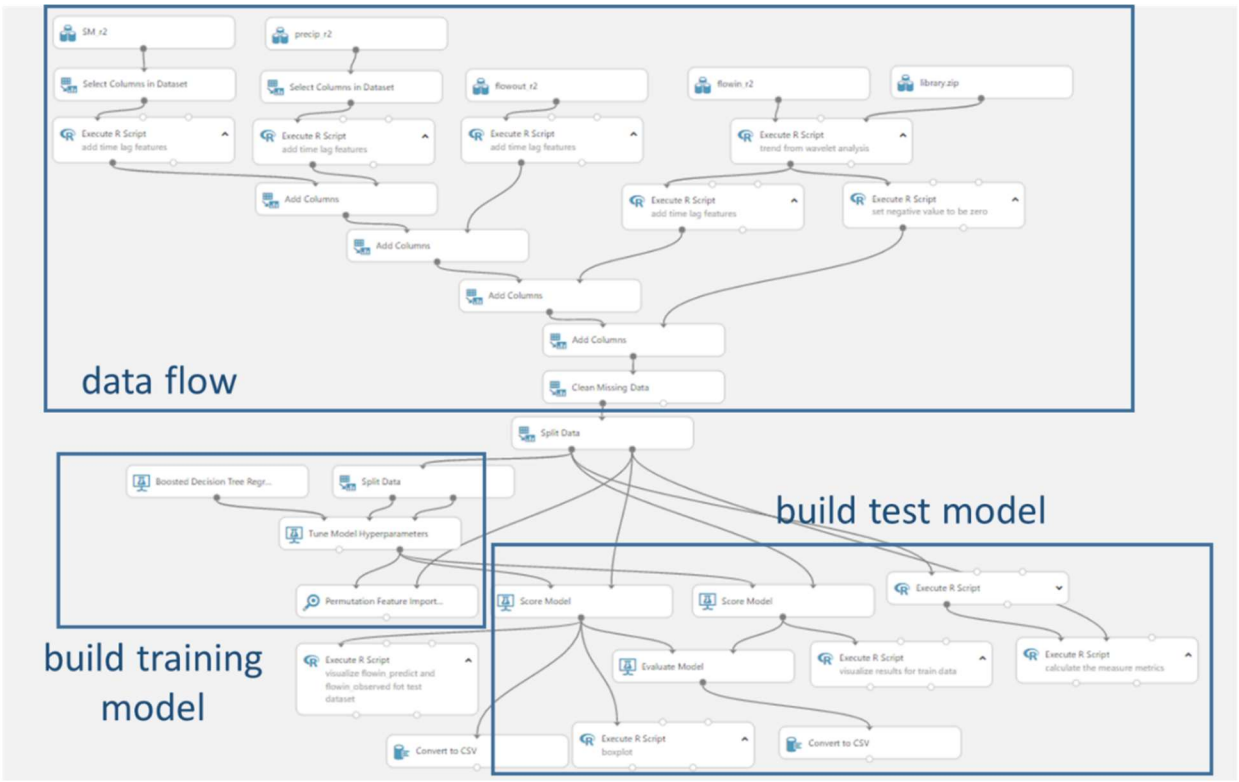

Figure 3.10. AzureML workflow for data-driven flow prediction model

A residual prediction model to predict residual between observed reservoir inflow and the predicted inflow from the physics-based model.

Figure 3.11 shows the plot of the residual errors, which were calculated as the filtered observed reservoir inflow minus the predicted reservoir inflow from the HEC-HMS model. HEC-HMS is a lumped parameter watershed model that simulates watershed response to precipitation and predicts flows throughout the watershed, including reservoir inflows [*Hydrologic Engineering Center*, 2011]. Based on the flow information, LCRA staff simulate reservoir operation using the HEC Reservoir System Simulation (HEC-ResSim) in CWMS, assess the impacts of the operations using HEC Flood Impact Analysis (HEC-FIA), and make decisions for reservoir management (e.g., determine reservoir releases to meet reservoir and downstream operational goals). The same input features as the above flow prediction model were applied here. The seven experiments described above were repeated for the hybrid model, with

the best-performing experiment selected.

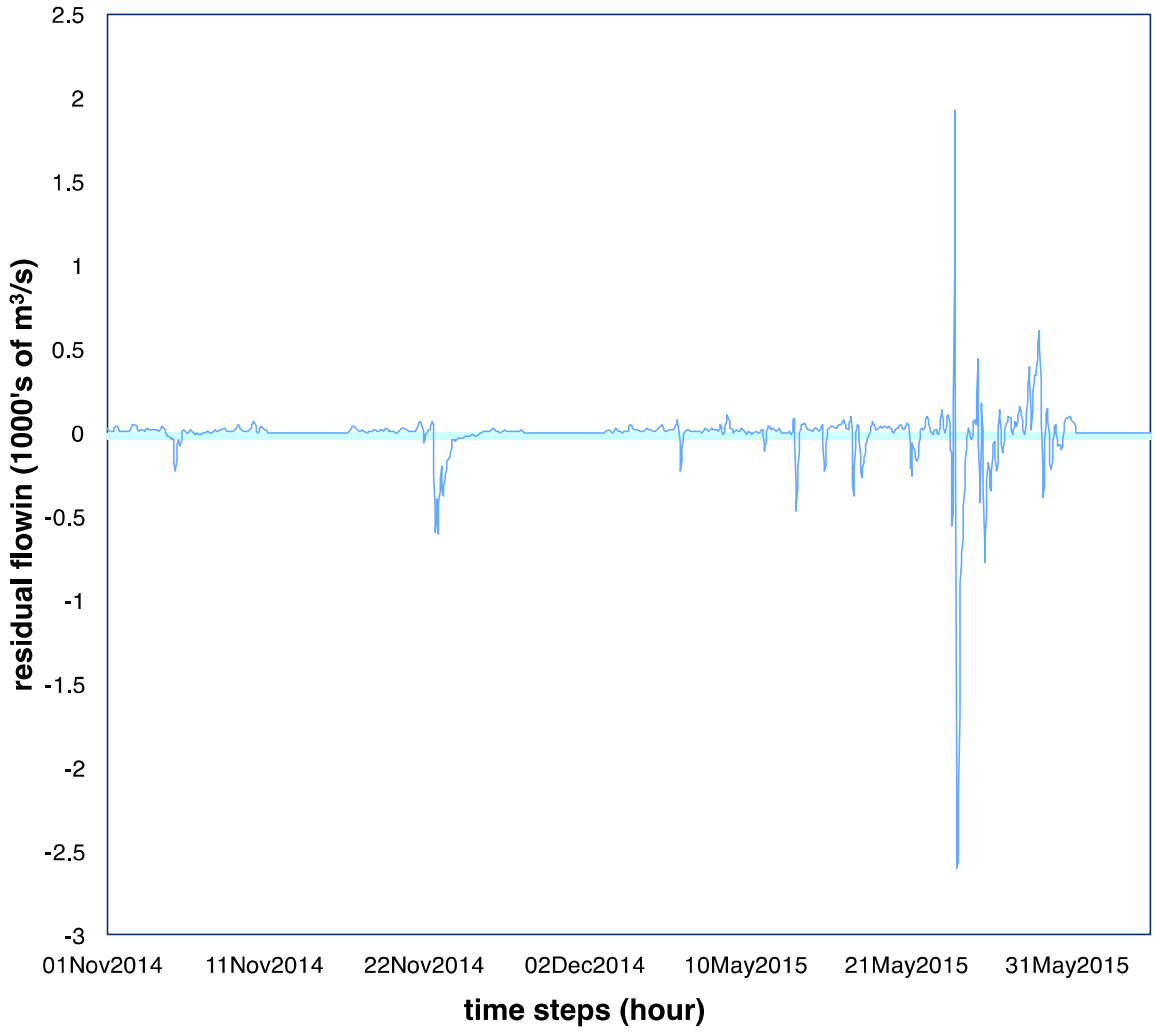

Figure 3.11. Residuals between filtered observed *flowin* and *flowin* from physics-based models during 2014–2015 flooding events

3.4 Results

3.4.1 Physics-based Model Performance

Figure 3.12 shows the predicted reservoir inflow from the physics-based model HEC-HMS in CWMS and Table 3.1 shows the performance metrics for the physics-based model. The results show that the physics-based model fits the general trend of the reservoir inflows but a residual error remains that can be fit with the hybrid model.

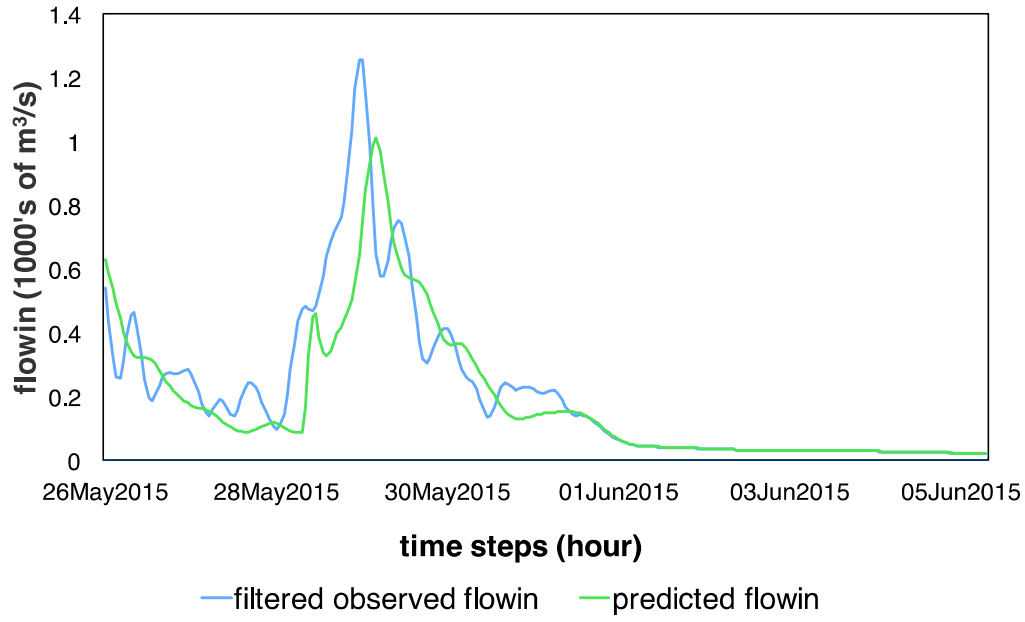

Figure 3.12. **Filtered** Observed *Flowin* vs Physics-based *Flowin* during 2014-2015 Flooding Events

Table 3.1. Performance Metrics of Physics-based Model

Mean Absolute Error (m^3/s)	**Root Mean Squared Error (m^3/s)**	**Relative Absolute Error**	**Relative Squared Error**	**Coefficient of Determination**
67.677	130.541	0.381	0.282	0.718

3.4.2 Data-driven Flow Prediction Model

Table 3.2 shows the performance of the data-driven flow prediction model for the seven experiments. Experiment #4 (soil moisture at time *t*-2 at the reservoir-located grid point, precipitation at time *t*-1 at the reservoir-located grid point, *flowout* at time *t*-2, and *flowin* at time *t*-1) demonstrates the best performance metrics. We can see that the flow prediction, shown in Figure 3.13, is close to the real reservoir inflow, with the prediction capturing both the general trend of the reservoir inflow and closely matching the peak values.

A comparison of experiment #1 and experiment #2 shows that the closest soil moisture estimate (experiment #2) is more effective than all 31 available estimates in the area (experiment #1), indicating that some input features are not improving predictions of reservoir inflow.

Experiments #2 through #5 demonstrate that a time lag of 2 hours for soil moisture input (experiment #4) is the best option, despite the correlation results showing a time lag of zero having maximum correlation. Experiment #7 has similar performance to that of experiment #6, possibly because the additional input variables in experiment #7 (precipitation at time *t*-2 and reservoir inflow at time *t*-2) provide trivial information to improve the prediction performance.

We also conduct experiments to predict reservoir inflow 1 to 9 hours ahead using the same input variables in Tables 3.2 and 3.3. Figure 3.14 shows the RMSE of future predictions from the data-driven flow prediction model. After 1 hour, the RMSE increases sharply, then fluctuates, indicating that while the flow prediction model can be used to predict reservoir inflow one hour ahead, later performance drops off significantly.

Table 3.2. Performance Metrics for Flow Predicted Model

	Input Variables	**Output Variable**	**Performance**				
			Mean Absolute Error(m^3/s)	**Root Mean Squared Error(m^3/s)**	**Relative Absolute Error**	**Relative Squared Error**	**Coefficient of Determination**
1	$SM(t)_{31}$, $Precip(t-1)_{31}$, flowout(t-2), flowin_lag(t-1)	flowin(t)	28.883	60.032	0.167	0.061	0.939
2	$SM(t)_{closest}$, $Precip(t-1)_{closest}$, flowout(t-2), flowin_lag(t-1)	flowin(t)	28.600	66.545	0.165	0.075	0.925
3	$SM(t-1)_{closest}$, $Precip(t-1)_{closest}$, flowout(t-2), flowin_lag(t-1)	flowin(t)	26.873	48.705	0.155	0.040	0.960
4	$SM(t-2)_{closest}$, $Precip(t-1)_{closest}$, flowout(t-2), flowin_lag(t-1)	flowin(t)	23.984	46.723	0.139	0.037	0.963
5	$SM(t-3)_{closest}$, $Precip(t-1)_{closest}$, flowout(t-2), flowin_lag(t-1)	flowin(t)	27.269	50.404	0.157	0.043	0.957
6	$SM(t)_{closest}$, $SM(t-1)_{closest}$, $SM(t-2)_{closest}$, $Precip(t-1)_{closest}$, flowout(t-2), flowin_lag(t-1)	flowin(t)	24.607	50.121	0.142	0.042	0.958
7	$SM(t)_{closest}$,$SM(t-1)_{closest}$,$SM(t-2)_{closest}$,$Precip(t)_{closest}$,$Precip(t-1)_{closest}$, $Precip(t-2)_{closest}$, flowout(t-2), flowin_lag(t-1), flowin_lag(t-2)	flowin(t)	27.666	52.953	0.160	0.048	0.953

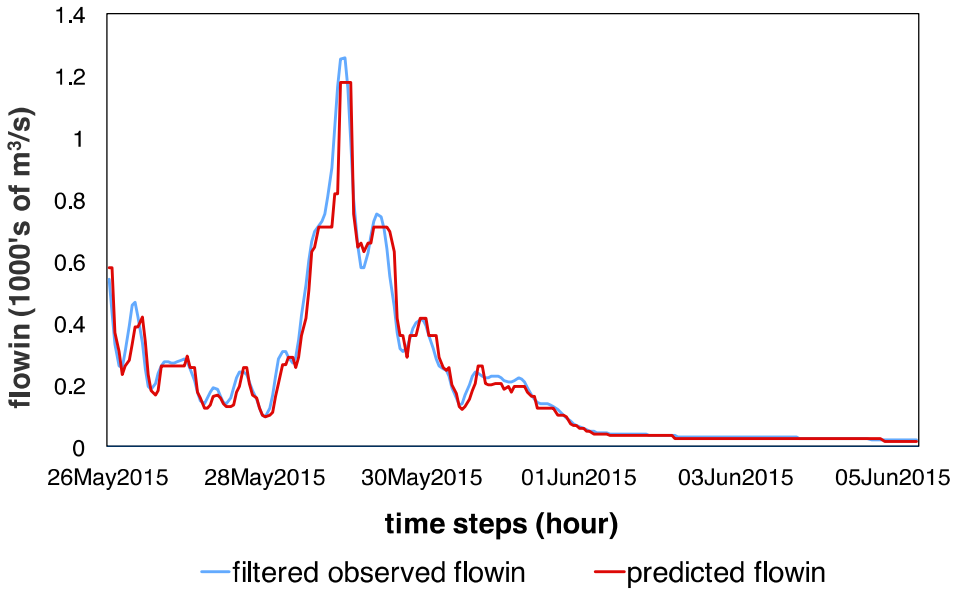

Figure 3.13. **Filtered** Observed *Flowin* vs Predicted *Flowin* from May 26, 2015 to Jun 5, 2015

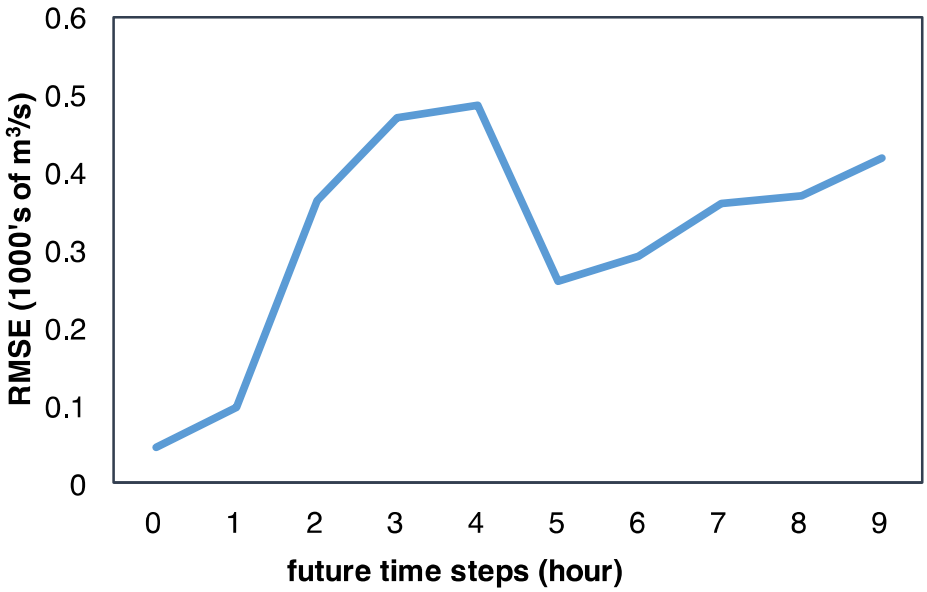

Figure 3.14. Flow Prediction Model Performance for Future Prediction

3.4.3 Residual Prediction Model

The hybrid prediction model is used to predict the residual error [residual(t)] between observed *flowin* and predicted reservoir inflow from the physics-based model, shown in Figure 3.11. The predicted reservoir inflow is then calculated using the predicted residual error plus the predicted reservoir inflow from the physics-based model. Table 3.3 summarizes the performance

of the hybrid model for each of the seven experiments, using the same input variables as for the flow prediction model. The best performance comes from experiment #2, followed by that of experiment #6. Since the hybrid model is intended to rapidly enhance the physics-based model's performance, it makes sense that the model including soil moisture has the best performance since CWMS does not consider soil moisture as an input [*Hydrologic Engineering Center*, 2011]. Figure 3.15 shows the performance of the physics-based model, the hybrid prediction model, and the observed *flowin* for Experiment #2. The hybrid prediction model improves upon the performance of the physics-based model in terms of the peak value prediction, but does not perform as well as the data-driven model in the short term (Figure 3.13).

Figure 3.16 shows the future prediction performance of the hybrid model. Within four hours, the RMSE curve fluctuates under 170 m^3/s. However, after four hours, the model's performance begins to drop off.

Table 3.3. Performance Metrics for Hybrid Prediction Model

	Input Variables	**Output Variable**	**Performance**				
			Mean Absolute Error (m^3/s)	**Root Mean Squared Error (m^3/s)**	**Relative Absolute Error**	**Relative Squared Error**	**Coefficient of Determination**
1	$SM(t)_{31}$, $Precip(t-1)_{31}$, flowout(t-2), flowin_lag(t-1)	residual(t)	81.836	167.636	0.472	0.475	0.525
2	$SM(t)_{closest}$, $Precip(t-1)_{closest}$, flowout(t-2), flowin_lag(t-1)	residual(t)	57.200	97.976	0.331	0.163	0.838
3	$SM(t-1)_{closest}$, $Precip(t-1)_{closest}$, flowout(t-2), flowin_lag(t-1)	residual(t)	71.925	121.196	0.415	0.249	0.751
4	$SM(t-2)_{closest}$, $Precip(t-1)_{closest}$, flowout(t-2), flowin_lag(t-1)	residual(t)	80.986	146.398	0.467	0.362	0.638
5	$SM(t-3)_{closest}$, $Precip(t-1)_{closest}$, flowout(t-2), flowin_lag(t-1)	residual(t)	69.943	138.186	0.404	0.322	0.678
6	$SM(t)_{closest}$, $SM(t-1)_{closest}$, $SM(t-2)_{closest}$, $Precip(t-1)_{closest}$, flowout(t-2), flowin_lag(t-1)	residual(t)	69.659	108.170	0.403	0.197	0.803
7	$SM(t)_{closest}$, $SM(t-1)_{closest}$, $SM(t-2)_{closest}$,$Precip(t)_{closest}$,$Precip(t-1)_{closest}$, $Precip(t-2)_{closest}$, flowout(t-2), flowin_lag(t-1), flowin_lag(t-2)	residual(t)	68.527	112.984	0.396	0.216	0.784

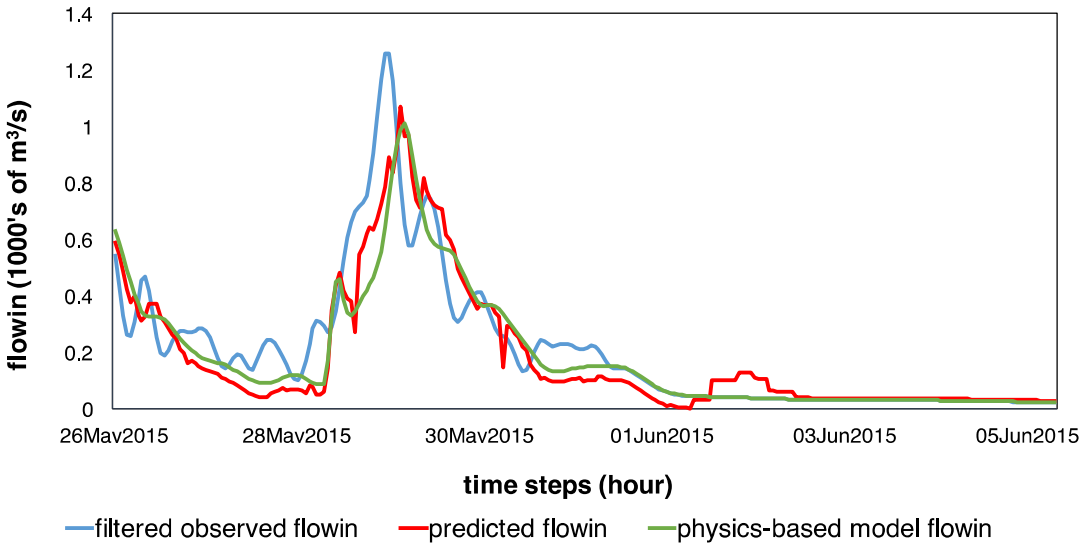

Figure 3.15. **Filtered** Observed *Flowin* vs Predicted *Flowin* from Hybrid Prediction Model vs Physics-based Model *Flowin* from May 26, 2015 to June 5, 2015

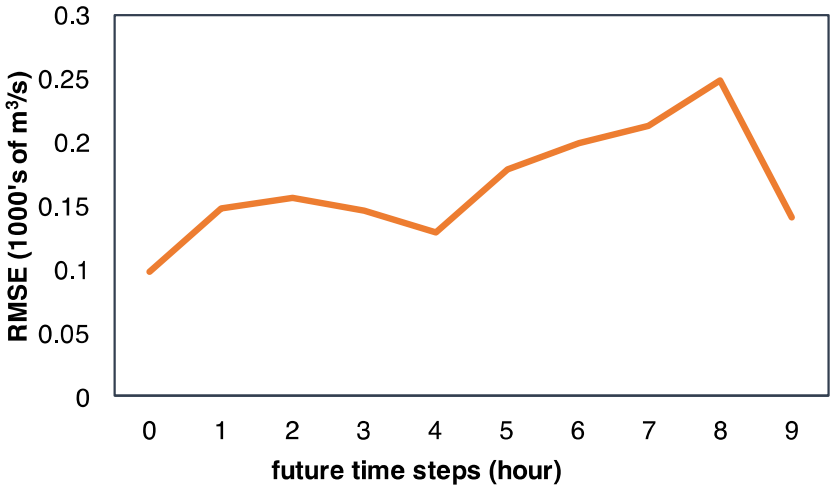

Figure 3.16. Hybrid Predicted Model Performance for Future Prediction

3.4.4 Web Interface

In AzureML, the built workflows were published as Web services using "Set Up Web Service" function. The Uniform Resource Locator (URL) and Application Programming Interface (API) Web Service keys were generated. The resulting data-driven services allow users and water managers to automatically fit model parameters, compute data-driven models, and retrieve reservoir inflow information through a Web browser. A Web application was built that enables users to give input parameters and retrieve output (Figure 3.17). Figure 3.17.a shows the user interface. The models can be executed in AzureML by filling the input parameter boxes and

selecting the "Compute" button; the result (the value of the predicted reservoir inflow) is retrieved and shown in the Web interface. The input parameters include the "StartTime" and "EndTime," which will automatically download precipitation and soil moisture from NLDAS2, as well as *flowout* (which is the flow exiting the upstream reservoir) and *flowin_lag* (which is the reservoir inflow in the previous time step).

Figure 3.17.b shows a prototype Web application that allows users to see the reservoir inflow prediction based on the prediction models. Users can provide a prediction starting time and a future prediction steps to examine how predictions compare with the measured data in the recent past, which will provide a sense for potential errors in the predicted reservoir inflows. In the future, when predicted soil moisture, precipitation and upstream reservoir *flowout* are available, such data can be incorporated into the prediction model to improve performance.

Furthermore, the Web application can easily be extended to other river basins. For instance, for any ungauged basin, users only need to upload the longitude and latitude of grid points affecting reservoir inflow to Azure ML. These points are then used to automatically download corresponding precipitation and soil moisture data from NLDAS2 using our workflow in AzureML. Then users can predict reservoir inflows based on the start time, end time, flowin_lag, and flow_out, as shown in Figure 3.17.a. Using this interface, the Web application provides an easy way for reservoir operators to forecast reservoir inflows and explore multiple scenarios without modeling or computational expertise.

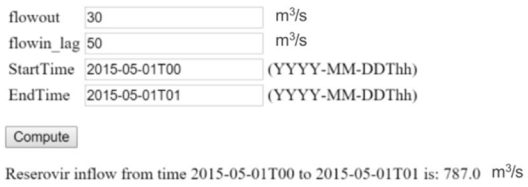

Figure 3.17.a. Flow Prediction Model to Calculate Reservoir Inflow

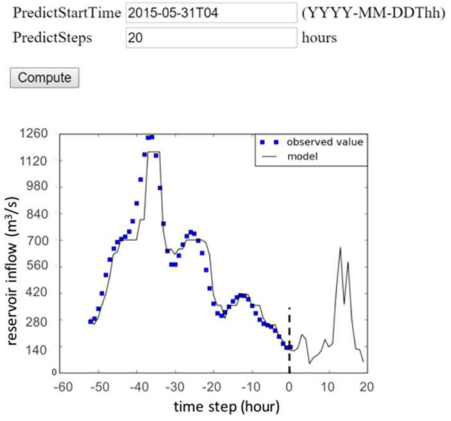

Figure 3.17.b. Reservoir Inflow Prediction

Figure 3.17. Screenshot of Web Interface for the Data-driven Model Services

3.5 Discussions

In this study, we propose a data-driven framework for real-time reservoir inflow prediction using a service-oriented approach that enables ease of access through a Web browser. Statistical and hybrid models are developed to predict flow and residual errors from a physics-based model, respectively. We created a workflow in Microsoft AzureML, a machine learning studio, for end-to-end downloading of the data, executing the models, and visualizing the results. Azure ML provides fast and easy implementation of the whole workflow as well as publishing of the workflow as Web services. In addition, the input datasets and workflow can easily be updated when new data are available. One of the workflows that predicts reservoir inflow has been published at *https://gallery.cortanaintelligence.com/Experiment/Predict-Reservoir-Inflow-1*. Users who wants to use AzureML to predict reservoir inflow can update the input data and the model will be automatically updated without manual calibration or tuning of model parameters.

The framework was implemented and tested in the Lower Colorado River Basin. The results show that the statistical flow prediction model is more accurate for short-term forecasts than the hybrid prediction model, while the hybrid model performs better for longer-term prediction (2 hours or more), as it considers forecasts from a physics-based model.

The flow prediction model has a peak prediction value close to the actual value. Of the set of experiments shown in Table 3.2, experiment #4 has the best performance. Using soil moisture at time t-2 at the reservoir-located grid point, precipitation at time t-1 at the reservoir-located grid point, *flowout* at time t-2, and *flowin* at time t-1 will lead to the best prediction of *flowin* at time t.

From a physical process perspective, soil moisture affects surface runoff by reducing infiltration. When flooding happens, infiltration has reached a saturated level. Therefore, high soil moisture conditions are indicative of wet conditions that are well correlated with high reservoir inflows and are thus useful for prediction.

The hybrid prediction model improves upon the performance of the physics-based model. Based on the set of experiments shown in Table 3.3, experiment 2 gives the best performance. Using soil moisture at time t at the reservoir-located grid point, precipitation at time t-1 at the reservoir-located grid point, *flowout* at time t-2, and *flowin* at time t-1 will lead to the best prediction in *flowin* at time t. The hybrid model's short-term performance is worse than that of the *flowin* prediction model. The hybrid model is affected by complex processes, as shown by

the high fluctuations in Figure 3.11, and available data to build the model are limited to just two flooding events. With more flooding events available in the future, the incorporation of more data will likely improve the model's performance.

In considering longer-term predictions, the hybrid prediction model is better than the data-driven flow prediction model in terms of RMSE (Figure 3.18). The flow prediction model's RMSE is lower than that of the hybrid prediction model one hour ahead. Later, the flow prediction model's RMSE is higher than that of the hybrid prediction model, indicating that the flow prediction model's performance declines after two hours. Because the hybrid prediction model's performance remains reasonably high within the following five hours, in the future the Web application could allow the user to create a combined prediction model that uses the data-driven model for the first two hours and the hybrid prediction model for time steps further in the future.

Further research is needed to explore how these findings generalize to other locations and storms. The models and tools developed in this work can easily be generalized to other reservoirs by updating the input data in the workflow. The workflow can also be combined with other modeling services requesting the Web service using URL and API keys, as mentioned previously.

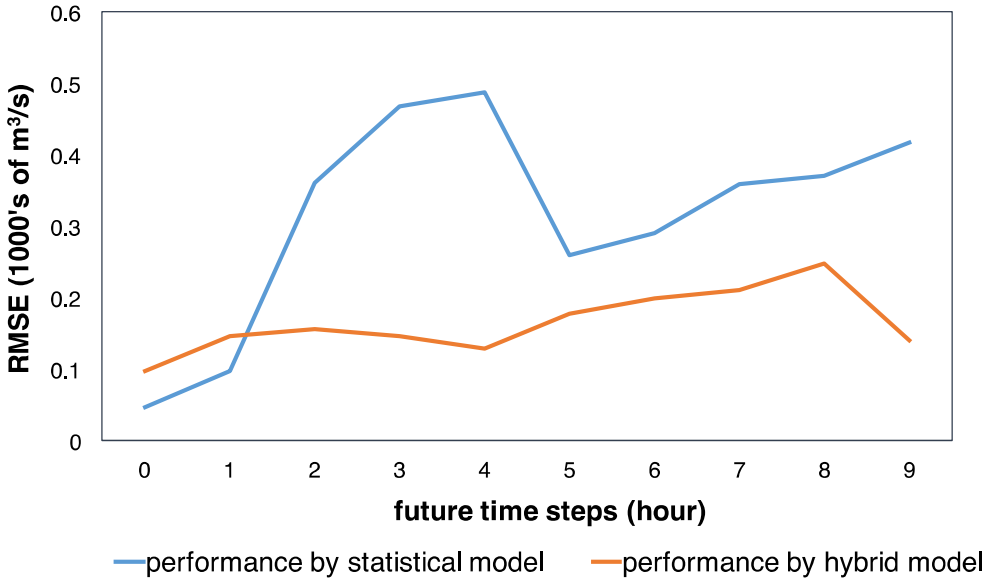

Figure 3.18. Prediction Performance of Flow Prediction Model and Hybrid Prediction Model

In the future, the hybrid prediction model for long-term prediction will need to be improved. Currently the only available CWMS forecasts from the LCRA database were *nowcasts* (forecasts for the current time period only). If longer-term CWMS predictions could be obtained, then the hybrid model might perform better for longer-term forecasts.

In addition, the current Web application is a prototype and further user-centered design and development is necessary before the system should be adopted for operational reservoir management. Feedback from LCRA's testing and evaluation of the Web application can be used to improve the interface and add more features as needed to support effective decision making. Moreover, when more flooding data are available, the data-driven and hybrid models can readily be updated and improved using the AzureML framework. Replacing historical data for soil moisture, precipitation, and upstream reservoir *flowout* with model predictions might improve reservoir inflow prediction in later time periods. For instance, the precipitation might be replaced by the Quantitate Precipitation Forecast (QPF) or local LCRA rain gauge data. In the future, other data preprocessing approaches such as partial information approach [*Sharma & Mehroma*, 2014, *Sharama, et al.,* 2016] could also be implemented to automatically choose the best input parameters for data-driven models to improve reservoir inflow forecast.

The findings clearly indicate promise for this type of approach and potential value in making datasets and model forecasts more readily available in real time to support such analyses. In addition to reservoir inflow forecasting, the framework can be extended to other water resources applications with rich data sets using the AzureML framework.

Chapter 4

A Service-driven Approach to Managing Water Allocation in Priority Doctrine Regions

Chapter 3 has demonstrated the application of a service-driven approach to predicting reservoir inflows during flood events. This chapter will explore a general framework of a service-driven approach to managing water allocation in Priority Doctrine Regions (Section 4.2) and its application to a drought event in the Upper Guadalupe River Basin (Section 4.3). Results and discussion are presented in Sections 4.4 and 4.5.

4.1 Introduction

As discussed in Chapter 1, water shortages originating from the imblance of water supply and water demand have had various social and economic impacts and raised water allocation issues for decision makers. As the limitations of water resources become more severe with growing populations and changing climate, it is challenging to make water allocations in a fair manner that improve sustainability of ecosystems. Decision makers commonly rely on water rights to determine the allocation of water to each stakeholder. There are two types of water rights in the US: riparian rights, which are based on land ownership, and prior appropriation rights, which are based on the rule of priority: "first in time, first in right."

Riparian water rights are common in the east. Most western states, which are naturally drier, follow the prior appropriation doctrine. The Guadalupe River Basin in Texas adopts the prior appropriation doctrine (called Priority Doctrine in this work), where water usage priority is based on the date of first diversion. During water shortages, longest-term appropriators (senior water users) have priority over shortest-term appropriators (junior water users) to receive water [*Huffaker et al*., 2000].

This study's purpose is to develop an optimization model for determining optimal water allocation strategies under a daily drought scenario in a prior appropriation doctrine system. A river forecasting model, which provides information about the amount of water in the river system, is coupled with an optimizaiton model for decision support. A real-time Web application is then developed to improve the real-time water allocation process by automating collection of information and coupling the river forecasting and optimization models through a service-oriented approach. The framework can easily be deployed in any areas that have a priority

doctrine policy for water allocation by updating water user information and river data.

In this study, we develop river modeling services, accessible through a user-friendly Web application, that execute on a National Center for Supercomputing Application (NCSA) server. Implementing river forecasting models as a model service can allow non-technical users to predict river streamflow and retrieve results directly online. Optimization services are then deployed in Azure Machine Learning (AzureML), which can easily call the river model services to support the decision-making process without manual model configuration or software installation. Decision makers can utilize the system to allocate water through a more scientific approach to assist subjective judgment.

The built coupled model services are applied to the Upper Guadalupe River Basin, Texas, during a serious drought event, focusing on water allocation on April 1, 2015, as a case study. Scenario analyses, including single scenario analysis and robust scenario analysis, are conducted to address uncertainties that affect water management and to inform future water supply and management.

4.2 Methodology

Figure 4.1 shows the entire water allocation framework, which consists of three major components: 1) River modeling service, which is the hydrological simulation model service deployed through the NCSA-developed Datawolf workflow tool; 2) a GA optimizer deployed through AzureML; and 3) a Web application to provide browser access. The GA optimizer in AzureML provides initial conditions such as river runoff and water diversion to the river modeling services and retrieves river streamflow predictions used by the optimization model. After both services are built, a Web application can be developed to provide ease of access to the model services for water management. Users can give input data and parameters through a Web browser and visualize the result via the Web interface. The following subsections discuss each component in more detail.

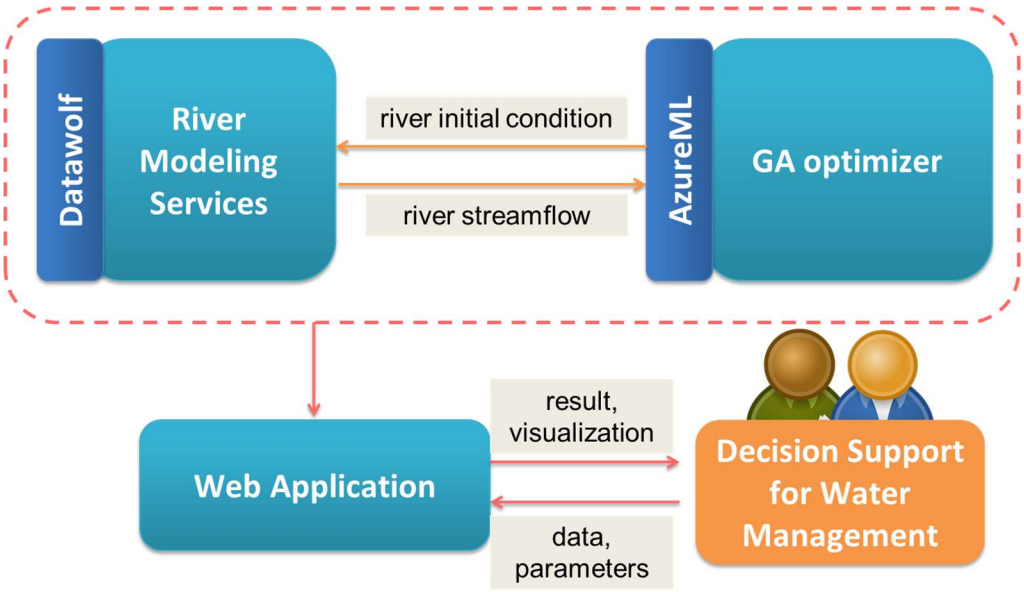

Figure 4.1. Framework for Decision Support Services during Drought Events

4.2.1 River Modeling Service

Villa et al. [2009] explained that a model application can be divided into several disciplinary steps and deployed as an automated pathway using a workflow approach. Workflow systems can integrate different components into an automation system in a loosely coupled environment [*Georgakopoulos et al*., 1995]. Workflows in which components do not affect one another can overcome technology-based software barriers such as different data standards, programming languages, and model compliers for each component.

DataWolf is an NCSA-developed workflow system for generating and publishing workflows as services. It has advantages over other workflow tools such as Taverna [*Oinn et al.*, 2004] and bio-STEER [*Lee et al.*, 2007] in wrapping modules as Web services without requiring programming skills [*Bajcsy et al.,* 2005]. Furthermore, DataWolf uses the OGC standards for storing data, which allows it to easily extend to new types of data [*Marini et al.,* 2007]. The DataWolf workflow tool has been published as a Web-based tool. Each user can build and access workflow tools through a Web browser.

Executing a large-scale hydrologic simulation model requires model configuration and user programming expertise, as well as requiring users to download data and software to execute models. DataWolf can use Python or a command line tool to build workflows for each step with specified inputs and outputs. Users can also check the intermediate results for each step. To illustrate the workflow creation process, the following subsection provides more details on the

RAPID model, which is a large-scale hydrological simulation model used in this work. This is followed by a subsection discussing the implementation of RAPID as a workflow using DataWolf.

RAPID Simulation Model

The RAPID model is a river routing model developed at the University of Texas at Austin for parallel computation of river discharge [*David et al.,* 2011]. Given the river network and its connectivity information, and fed with predicted water inflows (i.e., runoff) into the river network, RAPID can be executed to compute river streamflow for any river network. River connectivity information is provided by NHDPlus, which describes all river networks and water bodies in the United States.

The flow calculation in RAPID is based on the matrix-based Muskingum method [*McCarthy,* 1938]. The two important parameters *k* (a storage constant with dimension of time) and *x* (a dimensionless weighting factor characterizing the relative influence of the inflow and the outflow on the volume of the reach) in the Muskingum method are calculated based on the work of Cunge [*Cunge et al.,* 1969]. The parameter *k* is a constant value with dimension of time, and *x* is a dimensionless weighting factor influenced by the inflow and outflow of one specific river reach [*David et al.,* 2011]. These parameters are calibrated using any USGS gauges located in the river basin.

The basic expression for calculating river streamflow is given in Equation (4.1) by *David et al.* [2011]:

$$Q_j(t+\Delta t) = C_{1j} \cdot \left[Q_j^{up}(t+\Delta t) + Q_j^e(t+\Delta t)\right] + C_{2j} \cdot \left[Q_j^{up}(t) + Q_j^e(t)\right] + C_{3j} \cdot Q_j(t) \quad (4.1)$$

where t is time, Q_j^{up} is the upstream flow, Q_j^e includes lateral flows to the river network (e.g., runoff and groundwater seepage), Q_j is the streamflow in the exiting river reach *j*. C_{1j}, C_{2j}, and C_{3j}, constant parameters, are computed using Equations (4.2), (4.3), and (4.4):

$$C_{1j} = \frac{\frac{\Delta t}{2} - k_j \cdot x_j}{k_j \cdot (1 - x_j) + \frac{\Delta t}{2}}, \quad (4.2) \qquad C_{2j} = \frac{\frac{\Delta t}{2} + k_j \cdot x_j}{k_j \cdot (1 - x_j) + \frac{\Delta t}{2}}, \quad (4.3)$$

$$C_{3j} = \frac{k_j \cdot (1 - x_j) - \frac{\Delta t}{2}}{k_j \cdot (1 - x_j) + \frac{\Delta t}{2}} \quad (4.4)$$

where k_j and x_j are parameters in the Muskingum method [*Cunge*, 1969].

Workflow System

The RAPID model application is divided into three steps. The first step is to download the data from external data sources and prepare it for the RAPID input file. The second step is to execute the RAPID model with the RAPID input files from the first step. The final step is to visualize the RAPID output.

For each step, DataWolf is used to build a workflow tool with input data and parameters. For example, the command line wizard shown in Figure 4.2 gives users instructions about how to build a command line tool in DataWolf to execute the RAPID model. The RAPID input parameters of start-year, start-day, end-year, and end-day are then set in the DataWolf interface. An executable shell script is manually prepared to run each step. Given an input zip file from an external data source, each step can be executed with the set parameters and an output file is generated. Figure 4.3 shows all the defined steps of the RAPID model application built in DataWolf.

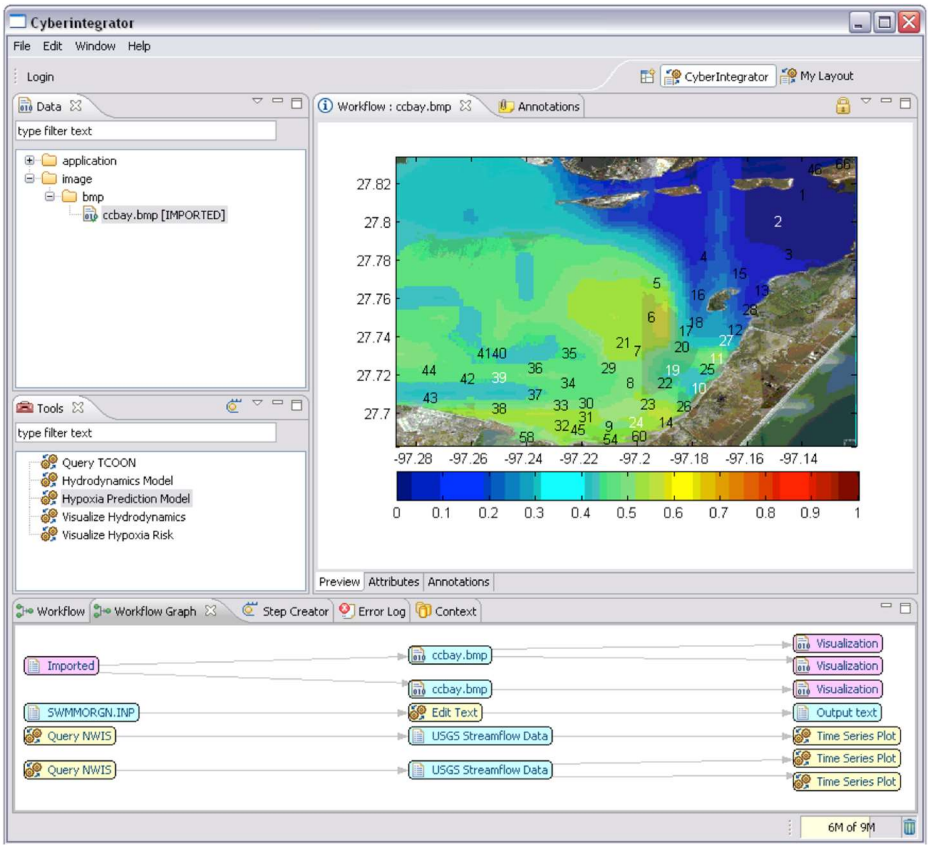

Figure 4.2. User Interface for DataWolf Workflow System

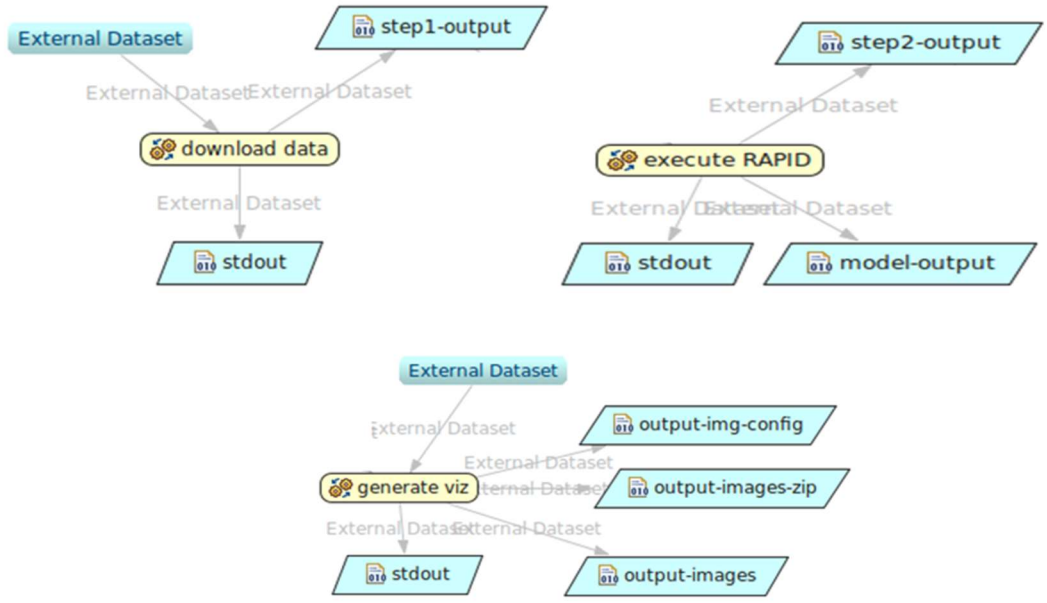

Figure 4.3. Defined Workflow Steps for RAPID Model Application

Once the three steps are implemented, DataWolf provides connecting tools to link each step. The output from the first step can be used as the input file for the second step. The output from the second step can be used as the input file for the third step, as shown in Figure 4.4 for the RAPID MaaS application.

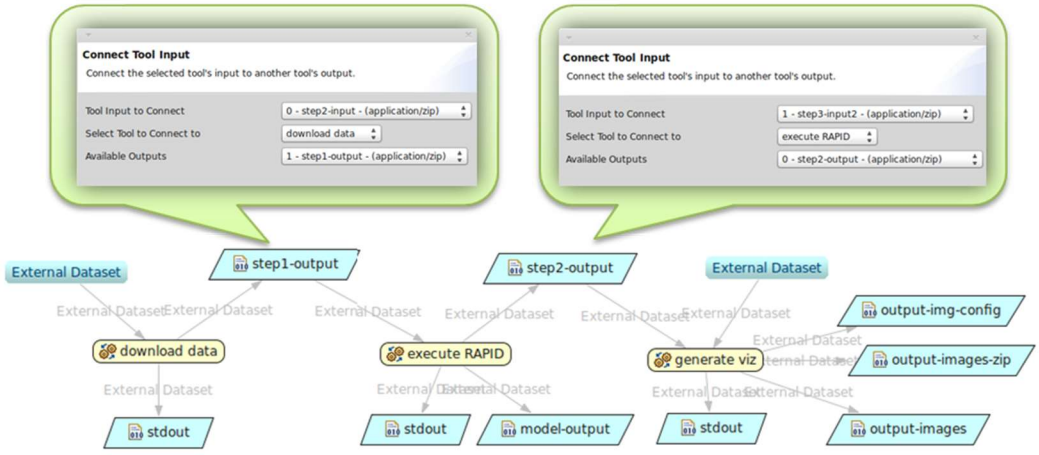

Figure 4.4. Connecting Tools in DataWolf that Link the RAPID MaaS Steps

The built workflow can be executed on a remote server or in the Cloud by providing input data and parameters that support supercomputing and parallel computing. Rather than maintaining a long-term interaction with the server or the Cloud during the model execution

process, users receive a notification when the model execution is finished. The final results can be downloaded or visualized through the DataWolf Web interface. In addition, the built workflow can be published as a Web service through the RESTful interface [*McHenry et al.,* 2011], providing access to the published workflows and their inputs and outputs through Uniform Resource Locators (URLs) that can be called as services remotely or through user sharing. This approach is general and can be applied with any river hydrology model.

After the workflow is built, users export the workflow as a zip file that is then uploaded to the DataWolf server. Users can access the shared workflow in the DataWolf server via the RESTful service using the URL of each workflow. For instance, clients can access the shared workflow and check on execution status using a Web browser by calling the URL. Through sharing the model and its output as services, the models become accessible to users anywhere in the world, which can dramatically expand their usability. Figure 4.5 shows the framework for publishing the workflow as a service.

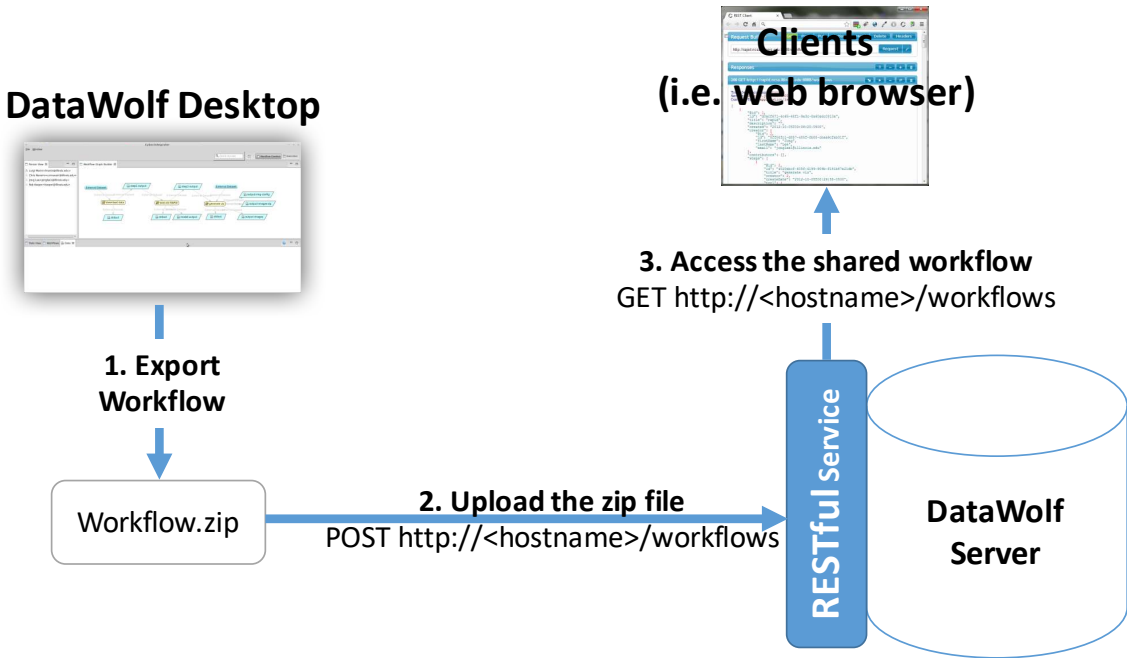

Figure 4.5. Publishing the Workflow as a Service

4.2.2 GA Optimizer

The GA module consists of optimization services built in AzureML, a workflow system for supporting machine learning services [*Azure ML team Microsoft*, 2015]. Users can provide

input data and parameters, run the optimization service in Microsoft Azure Cloud, and call the simulation model service on the remote NCSA server, viewing the results in AzureML.

Optimization Model

Instead of considering each water right permit holder individually in the optimization model, the water right permit holders are divided into groups to facilitate water allocation, following the approach of the Texas Commission on Environmental Quality (TCEQ), the regulatory organization that assisted with designing the case study. The objective of the optimization model is to minimize the daily total curtailment hours (the number of hours that each group of water right permit holders are not allowed to use water) across all groups of permit holders. This aims to reduce the total effects of water scarcity on each group of water users. There are two constraints in the formulation. The first constraint is to respect the priority doctrine, which says that senior water users have priority to withdraw water. This constraint ensures that the curtailment hours for the senior users will be smaller than those for the junior users. The second constraint is to ensure that the river streamflow after diversion still satisfies the minimum river streamflow requirement, which limits the amount of water that can be withdrawn from the river to maintain ecological sustainability.

The problem can be formulated as follows:

Minimize: the daily total curtailment hours in the whole river basin.

$$\min \sum_{i=1}^{N} x_i \quad (4.5)$$

Subject to the following constraints:

1) Respect the priority doctrine.

$$x_i \leq x_{i+1} \qquad i = 1, \dots, N-1 \quad (4.6)$$

2) Maintain the minimum river flow requirement.

$$Q(x_i, gauge) \geq Flow_{min} \quad (4.7)$$

where x_i represents curtailment hours for different groups of water users; *i* represents the group of water right holders; N represents the total number of water right holder groups in the Priority Doctrine; $Q(x_i, gauge)$ is the downstream river streamflow after water diversions in the river basin;$Flow_{min}$ represents the minimum river streamflow which is calculated based on the historical river flow recordings.

The second constraint as shown in Equation (4.7) is nonlinear and involves a series of complex equations (Equation (4.1) to Equation (4.4) in Section 4.2.1) that are solved

numerically. We use a GA as a metaheuristics optimization approach to solve this simulation-optimization problem.

Maier et al. [2014] pointed out that optimization as a service is a new direction for evolutionary algorithms because it facilitates linking multiple optimization and simulation models using workflows that users can easily access through Web applications. After the RAPID model service is built, the next step is to build the optimization model service and couple it with the RAPID service to construct simulation-optimization model services. In this study, the optimization service is built in Azure ML, a Cloud-based machine learning studio. The Azure ML python module executes the GA code by calling the RAPID service from the NCSA server and executing the whole framework in Microsoft Azure. Each Azure ML module is run on a single virtual machine (VM) in the Cloud. As shown in Figure 4.5, the input files generated in Azure ML are uploaded into the server where the RAPID model service is located. After RAPID has run, the output files are downloaded into the VM, where the optimization service can access them to check whether the constraint is satisfied and continue GA operations.

Genetic Algorithm

A real-coded Simple Genetic Algorithm (SGA) is implemented because the decision variables $(x_1, \dots, x_N)$, as shown in Equation (4.5), are real values. Figure 4.6 shows the SGA execution process. First, the population (a group of candidate solutions, or chromosomes) is initialized. Then, the fitness value (the objective function value) is calculated for each chromosome in the population. Through tournament selection, crossover, and mutation operators, the GA generates a new population, which serves as the next generation. The process repeats until the stopping criteria are satisfied.

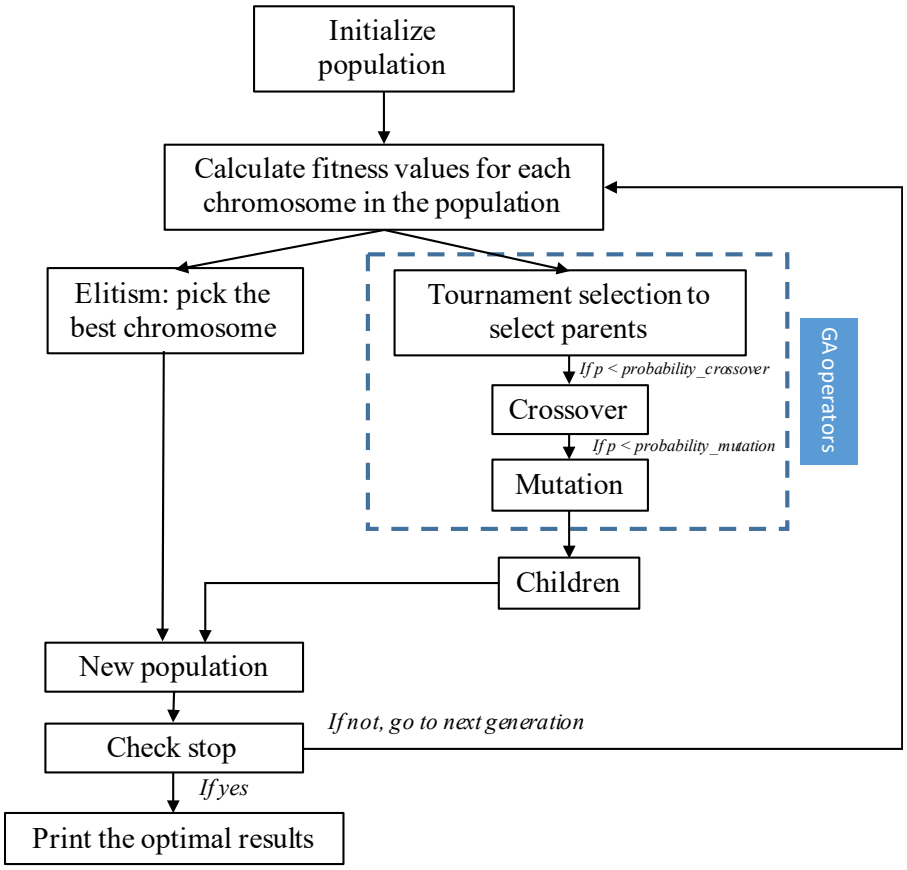

Figure 4.6. GA Execution Steps

- **Tournament Selection**

To handle the constraints in the constrained optimization problem, tournament selection is used to make pair-wise comparisons among chromosomes in the population [*Deb*, 2000]. Unlike penalty-based methods, which require tuning penalty weights [*Yeniay*, 2005], tournament selection searches for the true optimum solution by comparing feasible and infeasible solutions [*Deb*, 2000]. In this approach, if both candidate solutions satisfy the constraints, the one with the best objective function value is selected. If one solution is feasible and one is infeasible, then the feasible solution is selected. Lastly, if both are infeasible, then the one with the smaller constraint violation is selected.

- **Crossover and Mutation**

The crossover operator is a method of sharing information between two parent chromosomes (solutions) that assumes good chromosomes generate better offspring [*Herrera et al.*, 1998]. If a random value between zero and one is less than the predefined crossover

probability, a crossover operation generates new offspring. Otherwise, the parent chromosomes will pass to the next generation. The mutation operation explores the local region of a solution in the current population.

Figure 4.7 demonstrates the process of crossover and mutation. This study uses a simple two-point crossover for the GA crossover operator. Two positions are randomly chosen and children are generated by switching the genes of the parents between the two positions, as shown in Figure 4.7. The mutation operator increases the variability of the chromosomes. A random change is made based on mutation probability [*Herrera et al.,* 1998]. Figure 4.7 shows that a position is randomly chosen and the genes (one decision variable value) are randomly changed to generate a new chromosome.

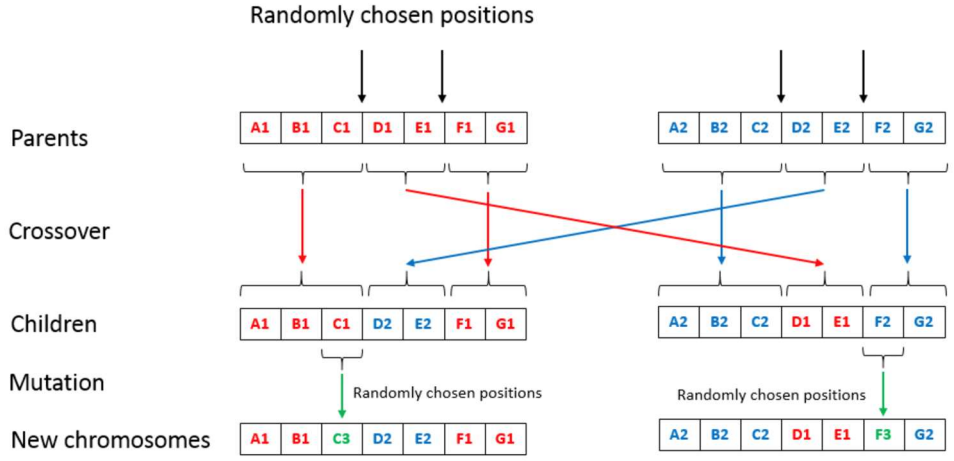

Figure 4.7. Crossover and Mutation Operators

- **Elitism**

Elitism passes the best chromosome in the current generation to the next generation. Retaining the best solution in each generation is intended to speed up the GA's convergence to the optimal solution [*Minsker*, 2005].

4.2.3 Web Application

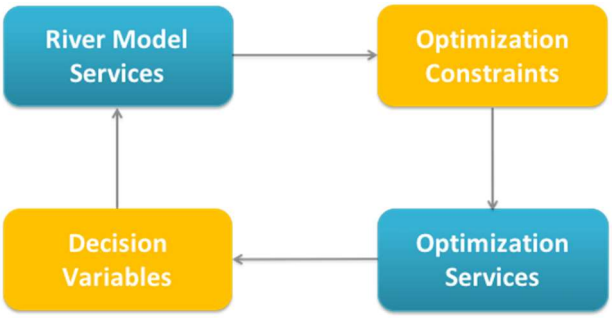

Figure 4.8. The Framework of Web Services

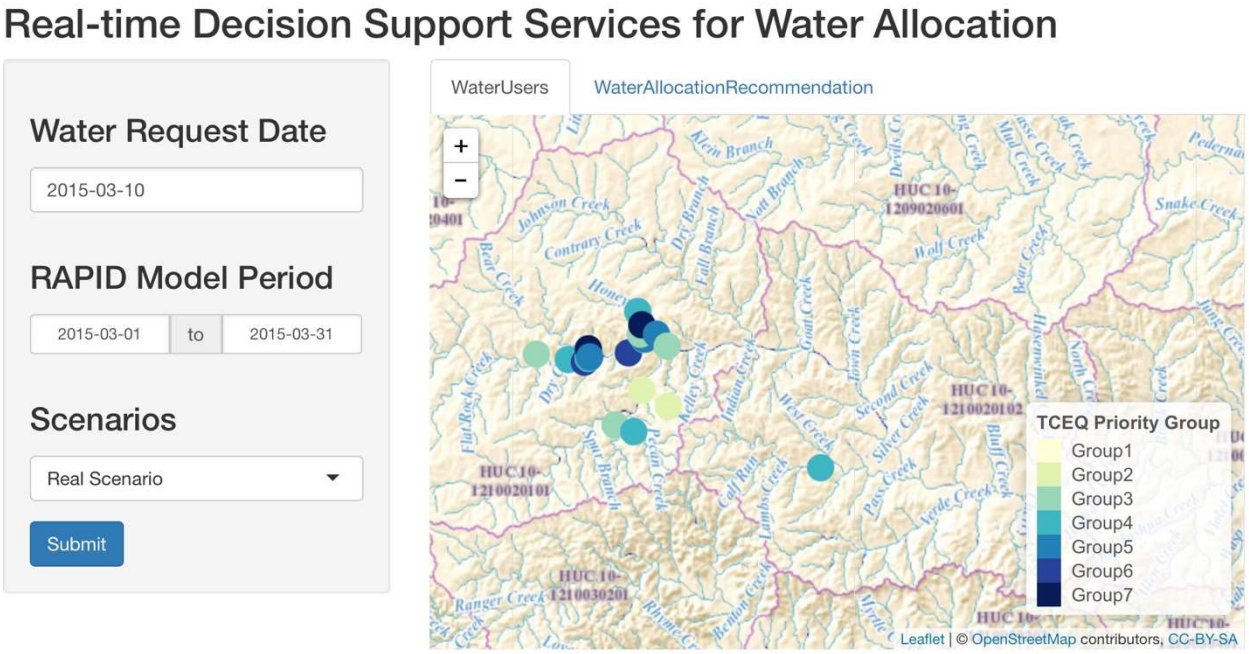

Figure 4.9. Web application for Real-time Decision Support Services

The decision variables from optimization services are fed into the river model service as one of the input files. The output of the river model service serves as optimization constraints for optimization services, as Figure 4.8 shows. Coupling the simulation model service and optimization service facilitates publishing the whole framework as a service and makes the models accessible to users through a Web browser. A Web application designed using R Shiny, a Web application framework based on R, is used for executing the simulation-optimization service.

The Web application retrieves water user information and visualizes the simulation-

optimization services (shown in Figure 4.9). This allows users to provide input parameters and execute the whole framework via a Web browser. Users can then retrieve and visualize the optimization results through the Web interface.

In addition, the scenario analysis panel allows users to analyze how the uncertainty of the runoff affects the future water strategy. Users can execute the simulation-optimization model and compare water allocation strategies under different scenarios, as described in the next section.

4.2.4 Scenario Analysis

Scenario analysis has been an efficient tool for decision support systems in water resources management, which explores future uncertainties to provide better decisions for water managers [*Pallotino et al.*, 2005]. The previous studies have used scenario analysis for uncertainties related to climate, population, political conditions, and other factors that can affect the water resources system performance [*Dong et al.,* 2013]. In this study, the uncertainties of weather and water demand are considered and scenario analyses of the alternative future states are explored.

Single Scenario Analysis

The single scenario analysis only considers one of the alternative future conditions. Given assumptions about future scenarios, the water allocation strategy under a single scenario will be analyzed to assist in water management. The simulation-optimization model services can assist decision makers in identifying the optimal water allocation strategy under each drought scenario. In addition, the optimization formulation in Section 4.2.2 shows that there are N groups of water right holders defined by water managers. We will explore the impact of this grouping on optimal water allocation strategies by setting different values of N.

The simulation-optimization system under a single scenario provides the optimal curtailment hours to each group of water right holders, assuming full compliance with the water restrictions. A few non-compliance scenarios are then proposed to examine how non-compliance would affect the river system and potentially violate the minimum river streamflow.

Robust Scenario Analysis

An alternative approach, described as robust scenario analysis, explores multiple scenarios simultaneously. It is intended to provide robust solutions that are flexible and satisfy various uncertain conditions [*Kang & Lansey*, 2013]. Some previous studies have focused on developing

robust solutions over a range of scenarios defined by uncertainties. Watkins & McKinney [1997] proposed a robust optimization framework that converges to a near optimal solution while satisfying all the stakeholder requirements across scenarios. Hamarat et al. [2014] developed a robust optimization approach based on a signal-to-noise ratio that is equal to the mean performance divided by its standard derivation. These previous studies focused on satisfying the feasibility of all the constraints but sacrificed the optimality of the solution. Since the scenario analysis is mainly reflected in the constraint as shown in Equation (4.10), the robust scenario analysis implemented in this work provides flexibility in constraint satisfaction. Given a user-specified probability that all scenario-related constraints are satisfied (e.g., 90%), the constraint is relaxed and the feasible region is extended to explore more alternatives without excessively risking river streamflow.

The robust optimization is formulated as shown in Equations (4.8) to (4.10). Equation (4.10) represents the probability that the satisfaction of all scenario-related constraints should be larger than the defined acceptance level of 90%.

The objective function is:

$$\min \sum_{i=1}^{N} x_i \quad (4.8)$$

Subject to the following constraints:

$$x_i \leq x_{i+1} \qquad i = 1, \dots, N-1 \quad (4.9)$$

$$prob(Q(x_i, gauge) \geq Flow_{min}) > 90\% \ \forall s \quad s = 1, \dots, S \quad (4.10)$$

where x_i represents curtailment hours for different groups of water users; S represents the total number of scenarios; s represents each single scenario; other parameters are the same as discussed in section 4.2.2.

4.3 Case Study

TCEQ employs water masters in certain river basins to provide active water management, particularly during droughts. The Priority Doctrine serves as a foundation for TCEQ's water management policies. Domestic and livestock users have priority water rights over any permitted surface water right holders. The permitted surface water right holders consist of senior water right holders, who were granted early water rights, and junior water right holders, who have obtained water rights more recently. Senior water right holders have higher priority to withdraw water than junior water right holders, except when health and safety are involved. During water

shortages, if all authorized water users' needs cannot be satisfied, water right holders can call TCEQ to carry out water allocation based on the Priority Doctrine [*L'Oreal Stepney*, 2012]. During this process, the amount of water available in the river, both currently and in the next few weeks, is one critical factor in water allocation for TCEQ water masters.

To manage water allocation more efficiently, TCEQ has divided the water right permit holders into seven groups. Each group of permit holders shares the same curtailment hours. Currently, water right holders are required to call TCEQ to request the amount of water they wish to use. During a severe drought, daily calls are required. After receiving a water request, TCEQ decision makers will check the amount of available water in the river by collecting information from USGS streamflow gauges. Then they allocate available water to each water user, respecting water users' priority, based on their subjective judgment as to the impacts of these allocations. The process is repeated each day as the impacts of the previous days' allocations become apparent in the river levels, but no forecasting is currently used. This process requires a large amount of time and the response of TCEQ decision makers is not immediate, nor is it based on the best available scientific forecasts.

This section describes this case study in more detail. The case study region is introduced in Section 4.3.1 and the framework of simulation-optimization model services is described in Section 4.3.2. Sections 4.3.3 and 4.3.4 discuss the scenario analysis implementation of the framework and Web application.

4.3.1 Case Study Area

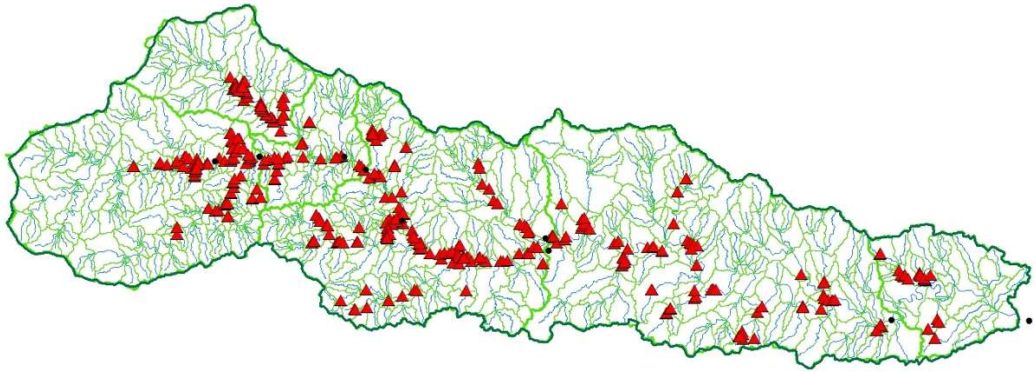

Figure 4.10. The Case Study Area

The case study area is the Upper Guadalupe River basin, located in the upper side of the Guadalupe River basin and 1432 mi^2 from the Canyon reservoir [*Bumgarner & Thompson*, 2012]. The climate pattern in this area is subtropical and subhumid, and the population in the largest city, Kerrville, Texas, was 22347 in 2010. The region's characteristic weather has made water allocation during water shortages a crucial issue in this area. Figure 4.10 shows the case study river basins. The red points in Figure 4.10 are the TCEQ water diversion points. The black points represent the USGS gauges.

4.3.2 Coupled Simulation-Optimization Services

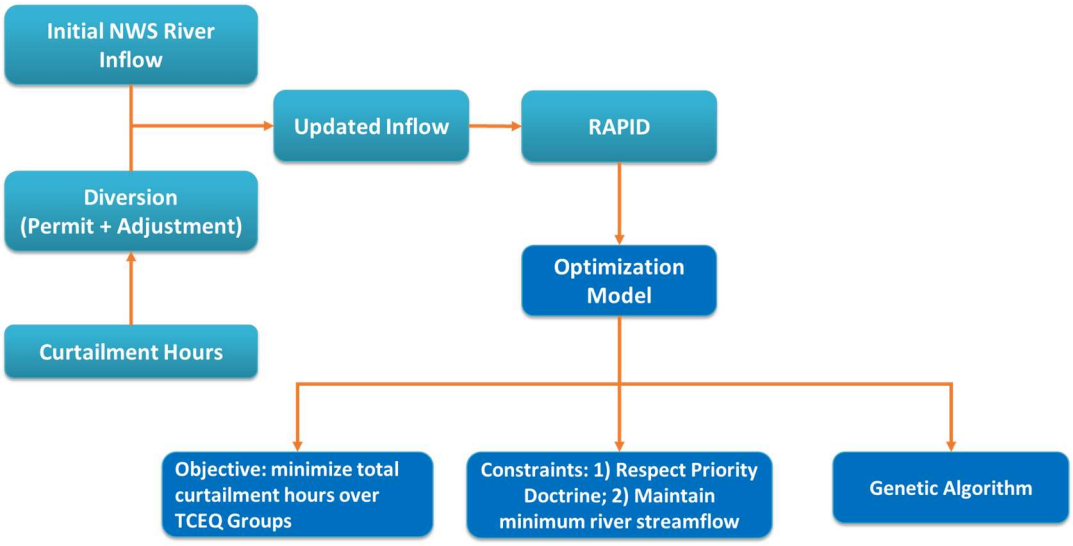

Figure 4.11. Optimization Service coupled with RAPID Model Service

The simulation-optimization model service is implemented using a recent drought event, focusing on April 1, 2015 allocations. Figure 4.11 shows the framework of the coupled optimization and RAPID model service. The decision variables in the optimization model are the curtailment hours of each group of water right permit holders. Based on the water demand and curtailment hours, the water diversion is calculated. Then the river inflow is updated using the water diversion values and initial National Water Service (NWS) river inflow. The river streamflow calculated by the RAPID model is used in the optimization model to ensure that the streamflow at the outlet of the river basin sustains the minimum river streamflow. The objective of the optimization is to minimize the total curtailment hours across all groups of permit holders. The optimization formulations have been given in Equations (4.5) to (4.7) for single scenario analysis and Equations (4.8) to (4.10) for robust scenario analysis. Since the RAPID model is nonlinear and the coupled simulation-optimization model cannot be solved using mathematical derivation, a GA is used to obtain the optimal solution to the water allocation problems as described in section 4.2.2. The input files for both the simulation and optimization models, including RAPID input files, water request data, and permit holder information, are uploaded into Azure ML. The permit holder information is from TCEQ. Water request data are daily water requests for each TCEQ permit holder in the river basin. Runoff files, which consist of hourly runoff data from April 2014 to August 2015, are downloaded from NWS and uploaded into

Azure ML.

GA Parameters

Based on Minsker's [2005] guidelines for GA parameter settings using binary GA theory (comparable guidelines are not available for real-coded GAs), the preliminary GA parameters in the study were set as follows. The minimum population size is set to $N = 1.4l$, where l=7 is the chromosome length in this case. After an initial test of different population sizes at and above this value (such as 100, 150, 200), the population size is set to be 100, which provides a set of sufficiently diverse chromosomes without excessive computational burden. The mutation rate is recommended to be set to the range of 0.04 to 0.3 [*Wright,* 1991]. We tested performance with mutation rate = 0.1 and 0.2 and found that mutation rate of 0.2 gives the best performance. In addition, the GA terminates when the stopping criterion, when successive GA iterations no longer produce better results, is satisfied. The number of successive GA iterations, N_stop, was tested using values of 10 and 20. The experiments showed that N_stop = 20 performs best. Table 4.1 summarizes the GA parameters used in this study.

Table 4.1. GA Parameters

Population Size	Crossover Rate	Mutation Rate	N_Stop
100	0.9	0.2	20

Ecological Flow

Equation (4.7) in the optimization formulation constrains the river streamflow in the downstream to be larger or equal to the minimum river streamflow. Richter et al. [2003] demonstrated that sustaining minimum river streamflow is critical in maintaining biological diversity and river ecology. Martin et al. [2014] showed that the ecology of the river should couple with the natural pattern of streamflow. Therefore, the minimum river streamflow, aiming to preserve sustainable river ecology, is computed from the historical river streamflow record during a "natural flow" period when no dams were constructed.

Canyon Reservoir, which is at the downstream end of the Upper Guadalupe River Basin, was planned in the 1930s and completed in 1964. The daily USGS gauge streamflow data in the upper stream of Canyon Reservoir is available from the middle of 1922. Considering the data integrity and the effects of reservoirs, the period from 1925 to 1936 is defined as the "natural flow" period for establishing the constraint criterion. The streamflow of USGS gauge 08167500,

which is located in the downstream of the Upper Guadalupe River Basin and upper stream of Canyon Reservoir, is used to set the minimum allowable river streamflow in the Upper Guadalupe River Basin in Equation (4.7) to the minimum river flow experienced during the natural flow period. For instance, the April 1, 2015, water allocation is used as a case study. The minimum allowable river flow is calculated from the minimum streamflow on April 1st during the natural flow period.

4.3.3 Scenario Analysis

Scenario analysis is an efficient tool to deal with uncertainties that affect water resource management [*Dong et al*., 2013]. This study focuses on weather uncertainty that affects river inflows and leads to changes in river streamflow, conducts scenarios for multiple river inflows, and investigates strategies of water allocation under different scenarios.

The historical TCEQ water demand data from April 2014 to August 2015 are used as a reference for obtaining water demand scenarios. Figure 4.12 is the cumulative probability graph of water demand generated using the first quantile, median and third quantile from the historical TCEQ water demand data. Suppose the probability of water demand by each water user is uniformly distributed. Given a random value of probability over the range from 0 to 1, a corresponding water demand value is generated using the cumulative probability graph. To explore different levels of water demand, four water demand scenarios that include “very high,” “high,” “moderate,” and “very low” are examined. Water demand under the “very low” scenario is randomly selected from the 0 to 10% quantiles in the cumulative distribution function. Similarly, the “moderate” scenario is selected from 10 to 30% quantiles, “high” scenario from 30 to 60%, and “very high” scenario from 60% to 100% quantile, as shown in Figure 4.12. We generated three runoff scenarios to represent different runoff levels: the actual runoff from NWS; drier runoff, which is calculated as actual runoff minus 1%; and wetter runoff, which is calculated as actual runoff plus 1%. Since the actual runoff on April 1, 2015 is a highly dry day, the wetter and drier runoff scenarios represent plausible scenarios under drought conditions.

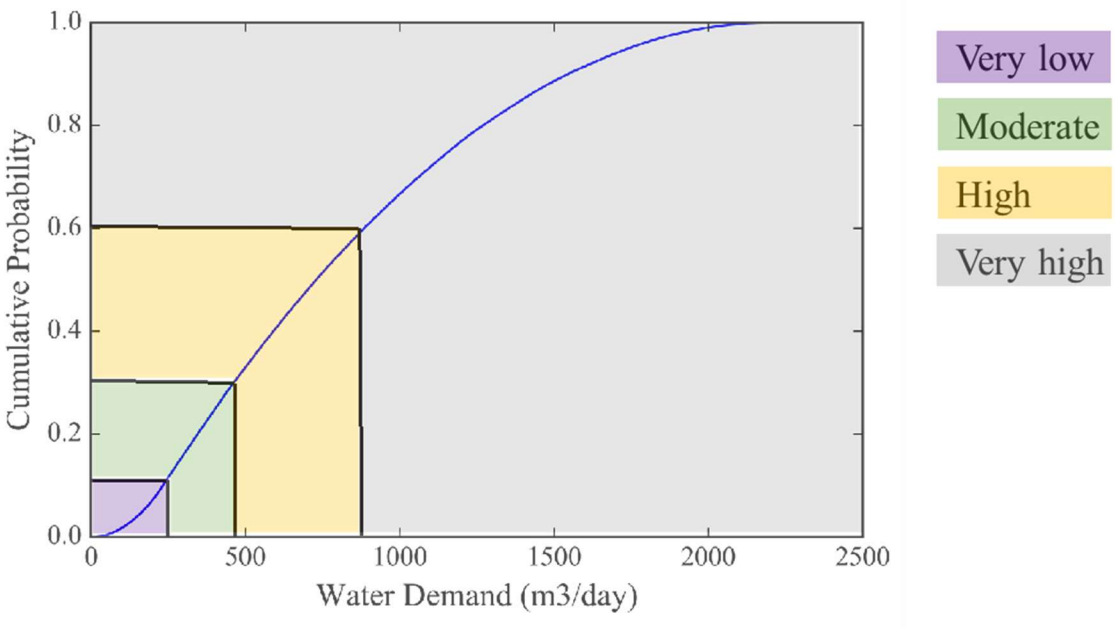

Figure 4.12. Water Demand Scenario Graph

4.3.4 Web Application

Figure 4.13 shows a prototype Web application that allows users to access the simulation-optimization services. Users need to set "optimization date", the date of the water request, and "simulation model execution period," the RAPID model execution period. The simulation-optimization model services are run in the Azure Cloud and NCSA server. The optimal water allocation strategy that presents the curtailment hours for each group of water right permit holders is retrieved via Web browser. Decision makers who are unfamiliar with the RAPID simulation model and the GA algorithm can easily use the Web application to aid subjective judgment in developing water allocation strategies.

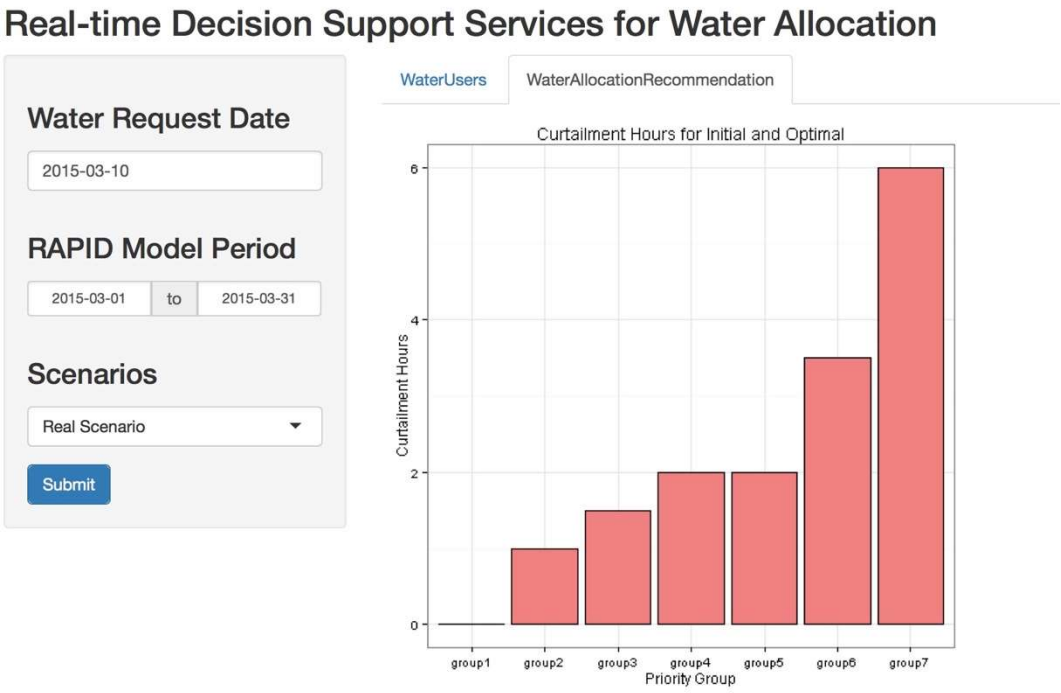

Figure 4.13. Web interface of Real-time Support Services for Water Allocation

4.4 Results

This section shows the results of implementing the built simulation-optimization model services for identifying optimal water allocation strategies under each scenario described in the previous section. In addition, the change in optimal water allocation strategies with different runoff levels and water demands is discussed. The USGS river streamflow data has shown that April 1st, 2015 was an extremely dry day in the Upper Guadalupe River Basin, thus it is selected as a case study. The drought scenario is developed from "drier" to "wetter" and the water demand scenario ranges from "very low" to "very high" as shown in Table 4.2. Table 4.3 shows all the scenarios by combining the uncertainty of water demand and runoff. The optimal water allocation strategy under each scenario, which is represented by the curtailment hours for each group of water right permit holders, is discussed below.

Table 4.2. Uncertain Variables

Uncertain Variables	Options
Water Demand (indicated by water request)	Very high
	High
	Moderate
	Very low

Table 4.2 (cont.)

Uncertain Variables	Options
Drought Situation (indicated by runoff)	Wetter
	Actual
	Drier

Table 4.3. Drought and Water Demand Scenarios

Scenario Name	Water Demand	Drought Situation
s1_actual	Very low	Actual
s2_actual	Moderate	
s3_actual	High	
s4_actual	Very high	
s1_drier	Very low	Drier
s2_drier	Moderate	
s3_drier	High	
s4_drier	Very high	
s1_wetter	Very low	Wetter
s2_wetter	Moderate	
s3_wetter	High	
s4_wetter	Very high	

4.4.1 Single Scenario Analysis

In this section, the optimal solution for each single scenario is identified using the built simulation-optimization model services. The GA is executed using three different random seeds and the best solution is chosen as the final solution for that scenario. The results of scenario analysis are discussed based on each runoff level. An alternative TCEQ priority grouping is proposed to explore the impacts of the grouping on water allocations. Then a few non-compliance scenarios, in which the water right permit holders do not obey the optimal water allocation strategy, are developed to analyze the compliance aspects of the water allocation strategy.

Table 4.4 and Figure 4.14 show the results of scenario analysis under the actual (historical) drought situation. The total curtailment hours of all groups increase from s1_actual to s4_actual due to the increasing water demand. The optimal curtailment hours for Groups 1, 2, 3, and 7 remain the same under different water demand scenarios. This indicates that the senior groups and the most junior group are not influenced by the water demand uncertainty. Water right permit holders in the most junior group (Group 7) are required to completely cut off water usage. Those in senior groups such as Groups 1, 2, and 3 can make full use of water. The curtailment hours for water right permit holders in Group 5 increase by 270% from s1_actual to

s2_actual. In the meantime, the curtailment hours in Group 4 decrease by 100%. This demonstrates that a decrease in curtailment hours for senior water right holders (Group 4) will lead to a sharp increase in the next group of more junior water right holders (Group 5). Group 6 also has a slight decrease (around 4%) in the curtailment hours from s1_actual to s2_actual, but this does not lead to an increase of curtailment hours in Group 7 because permit holders in Group 7 have been completely cut off. From s2_actual to s4_actual, the curtailment hours for Groups 5 and 6 increase significantly due to greater water demand, indicating that measures to reduce water demand (e.g., public education) could help permit holders in these groups significantly.

Table 4.4. Scenario Analysis Under Actual Runoff

	fitness	**Group 1**	**Group 2**	**Group 3**	**Group 4**	**Group 5**	**Group 6**	**Group 7**
s1_actual	56	0	0	0	2.5	6	23.5	24
s2_actual	68.5	0	0	0	0	22	22.5	24
s3_actual	71	0	0	0	0	23.5	23.5	24
s4_actual	71.5	0	0	0	0	23.5	24	24

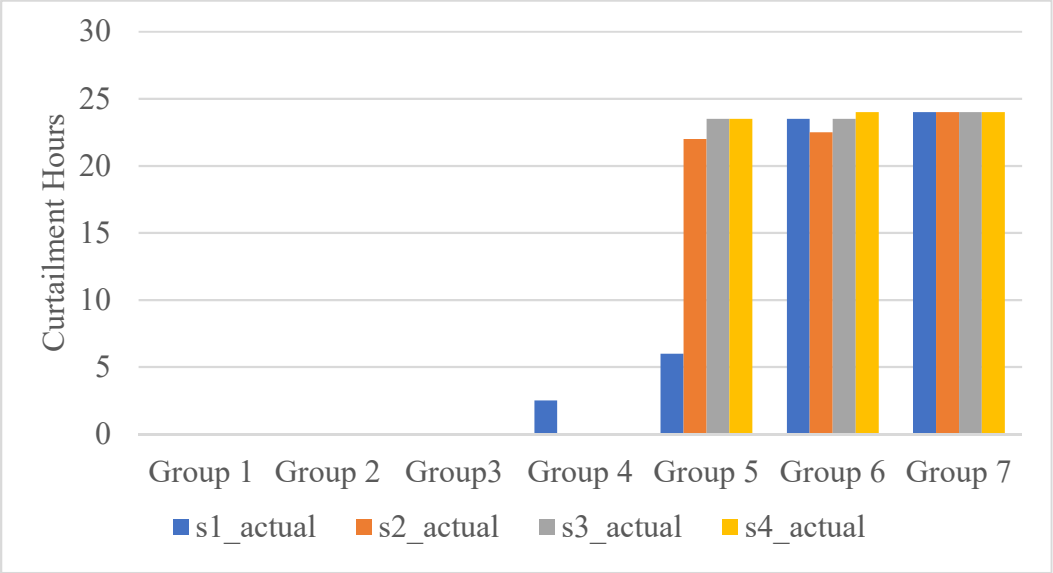

Figure 4.14. Scenario Analysis for Different Water Demand Levels under Actual Runoff

Table 4.5 and Figure 4.15 illustrate how the curtailment hours change for different water demand levels under a drier runoff situation. Compared to the actual runoff scenario where the senior groups, Groups 1, 2, and 3, can make full use of water, only the most senior group (Group 1) is guaranteed sufficient water usage. The curtailment hours for water right permit holders in Groups 2 and 3 increase by 65%, 30% and 5% from s1_drier to s4_drier. The junior groups such as Groups 5, 6, and 7 are required to completely cut off water usage during drier runoff

scenarios. In addition, when water demand increases from s2_drier to s3_drier, the curtailment hours in Group 4 decrease slightly (around 8%), but the curtailment hours in Group 5 are not altered since water right permit holders in Group 5 have been completely cut off.

These results indicate that the optimal water allocations are highly sensitive to small reductions in runoff (in this case 1%), having a greater effect on senior water users such as Groups 2, 3, and 4 compared to the actual situation. Under actual runoff, the dry conditions mainly affect water right holders in Groups 5, 6, and 7, while most senior groups can make full use of water. When the drought situation is worse, the curtailment extends to senior groups such as Groups 2, 3, and 4. Only the most senior group (Group 1) is not affected by the drought situation under the drier scenario.

Table 4.5. Scenario Analysis Under Drier Runoff

	Fitness	**Group 1**	**Group 2**	**Group 3**	**Group 4**	**Group 5**	**Group 6**	**Group 7**
s1_drier	102	0	10	10	10	24	24	24
s2_drier	129	0	16.5	16.5	24	24	24	24
s3_drier	137	0	21.5	21.5	22	24	24	24
s4_drier	141	0	22.5	22.5	24	24	24	24

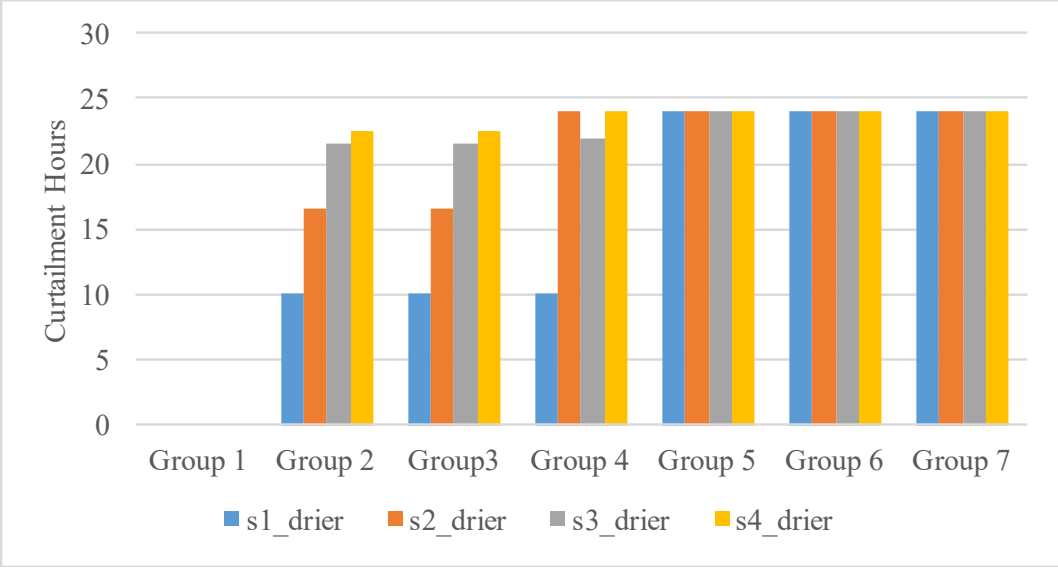

Figure 4.15. Scenario Analysis for Different Water Demand Levels under Drier Runoff

Table 4.6 and Figure 4.16 demonstrate how the curtailment hours change for different water demand scenarios under a wetter runoff scenario. Again, the results show significant changes for a small increase of 1% in runoff. The senior water right permit holders such as

Groups 1 and 2 can now make full use of water. Compared to the actual water scenario, water right holders in Groups 3 and 4 are required to slightly cut off water usage (0.5 -1 hours/day).

The decrease (around 67%) of curtailment hours in Group 4 from s2_wetter to s3_wetter leads to an increase (around 89%) of the curtailment hours in the following junior group (Group 5). Under s1_wetter, the curtailment hours in Group 6 decrease by 83% compared to the actual drought situation. From s1_wetter to s4 wetter, the curtailment hours in Group 5 increase by 260%, 89%, and 18%, and hours for Group 6 increase by 438%, 7% and 2%. The only scenario where water right permit holders in Group 7 can use water is s1_wetter. They are required to completely cut off water usage under other scenarios.

Therefore, the increase of available water in the system (wetter scenario) benefits Groups 5 and 6 more and has a slight effect on senior groups (Group 1, 2, 3, and 4) and the most junior group (Group 7).

Table 4.6. Scenario Analysis under Wetter Runoff

	Fitness	**Group 1**	**Group 2**	**Group 3**	**Group 4**	**Group 5**	**Group 6**	**Group 7**
s1_wetter	29	0	0	0	0.5	2.5	4	22
s2_wetter	56	0	0	0.5	1	9	21.5	24
s3_wetter	61.5	0	0	0	0.5	13	24	24
s4_wetter	67	0	0	0	0.5	18.5	24	24

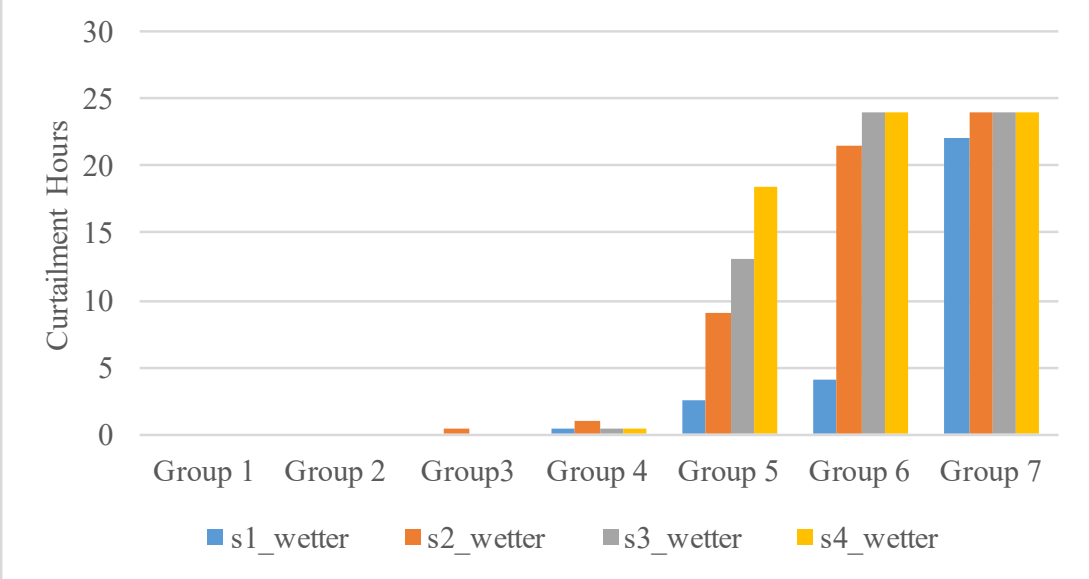

Figure 4.16. Scenario Analysis for Different Water Demand Levels under Wetter Runoff

Overall, the most senior water right holders (Group 1) and the most junior water right holders (Group 7) are insensitive to the uncertainty of runoff and water demand. The curtailment

hours for Group 1 over all single scenarios are zero and Group 7 is required to completely cut off water usage except in s1_wetter, which has the smallest water demand and wetter runoff. Figure 4.17 shows the sensitivity analysis of the curtailment hours for each group of water right permit holders under each runoff scenario with the uncertainty of water demand.

Under the actual runoff scenario, water right holders in Groups 1, 2, 3, and 7 are not influenced by the uncertainty of water demand. Group 5 is the most sensitive to the uncertainty of water demand, followed by Groups 4 and 6. When water shortages happen, the imbalance of water supply and water demand cannot fulfill all water users. The water demand of senior water users (Groups 1, 2, and 3) is satisfied first. Since the remaining water is not enough for other users, water users in the middle groups (Groups 4, 5, and 6) have their water demand partially fulfilled, and the most junior water users (Group 7) have no water to use.

A similar trend occurs in the drier and wetter runoff scenarios, as shown in Figures 4.17.b and 4.17.c. Groups 2, 3, and 4 are mainly affected by the uncertainty of water demand in the drier runoff scenario. In the wetter runoff scenario, Group 6, followed by Group 5, are the leading groups impacted by the uncertainty of water demand.

When water supply declines from the actual runoff scenario to the drier runoff scenario, the available water calculated by water supply minus water demand decreases, which only suffices for the most senior group's water consumption (Group 1). All the remaining groups lack sufficient water. The remaining water after withdrawal by Group 1 is not adequate for Groups 2, 3, and 4, while the junior groups (Groups 5, 6, and 7) are fully blocked from water usage. When the drought situation shifts from the actual runoff scenario to the wetter runoff scenario, the more junior groups become the most sensitive due to the increment of available water.

For those groups who are mainly affected by the uncertainty of water demand, TCEQ may encourage them to request conservative amounts of water during droughts, because smaller levels of water demand can increase the available water usage hours. In addition, water permit holders in more junior groups may find investments in water retention devices (e.g., water tanks) worthwhile to ease the fluctuations. Moreover, a water market for trading with the senior water users to satisfy water demand during droughts may be helpful, since senior water right permit holders are still eligible to withdraw water. Finally, the results illustrate the importance of accurate runoff estimation to enable informed water management. Even a small fluctuation of 1% in runoff volume has a dramatic impact on many users' optimal water allocation.

In addition, the optimal water allocation strategy for the highest water demand under the drier runoff scenario meets the streamflow constraint for all other scenarios since this scenario represents the worst drought condition. If extreme drought happens and water demand information is not available, TCEQ watermasters can allocate water based on this conservative strategy.

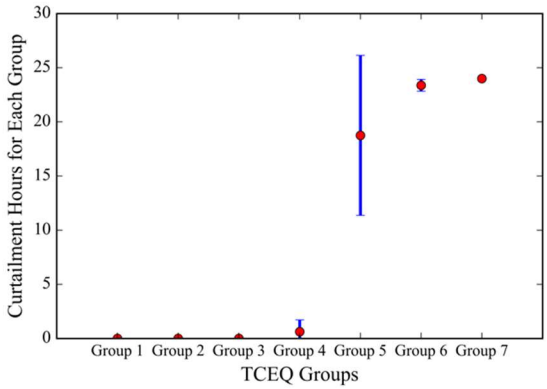

Figure 4.17.a. Actual Runoff Scenario with Uncertainty in Water Demand

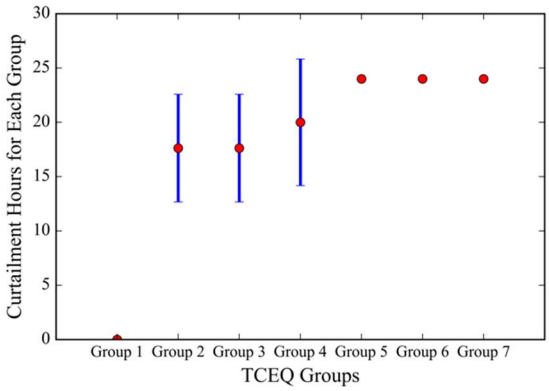

Figure 4.17.b. Drier Runoff Scenario with Uncertainty in Water Demand

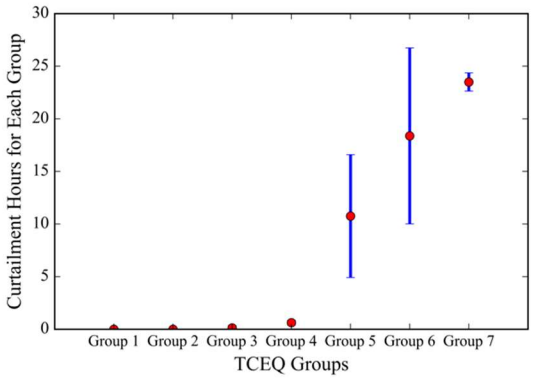

Figure 4.17.c. Wetter Runoff Scenario with Uncertainty in Water Demand

Figure 4.17. Average and Standard Deviation of Curtailment Hours for Each Group across Different Runoff Scenarios

Alternative TCEQ Priority Grouping

The previous single scenario analysis shows that water users in some groups under a scenario have the same curtailment hours. For instance, the water right permit holders in Groups 2 and 3 with different priority year have the same curtailment hours under the drier runoff scenario. The priority of each water user is based on the priority year and there are only seven groups of users in the current TCEQ priority grouping. If more groups are created among all

water users, the water allocation strategy may be fairer due to a smaller interval of priority years in each group. The objective of this section is to propose an alternative TCEQ priority grouping which uses more groups among all water users and to explore how the alternative grouping changes the optimal water allocation strategy.

Table 4.7. Alternative TCEQ Groups

Group	Original	Alternative
1	1887-1896	1887-1900
2	1900-1920	1900-1910
3	1920-1935	1910-1920
4	1935-1950	1920-1930
5	1950-1965	1930-1940
6	1965-1980	1940-1950
7	1980-now	1950-1960
8		1960-1970
9		1970-1980
10		1980-1990
11		1990-2000
12		2000 - now

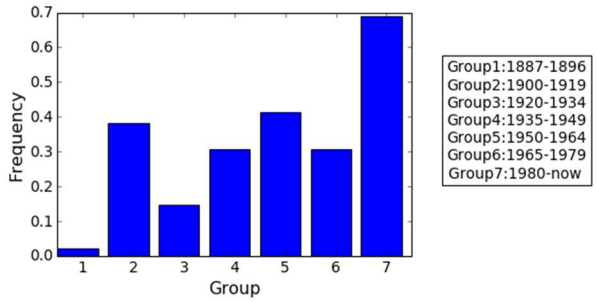

Figure 4.18.a. Original TCEQ Groups

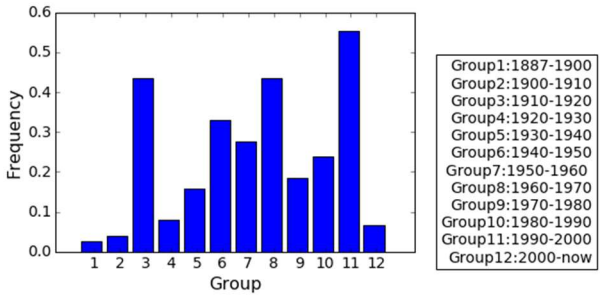

Figure 4.18.b. Alternative TCEQ Groups

Figure 4.18. TCEQ Group Information

Table 4.7 lists the original TCEQ groups and proposed TCEQ groups. Instead of using 15-year intervals from 1900 to 1980 in the original TCEQ priority grouping, the alternative TCEQ priority grouping still respects the priority year of each water right holder but uses 10-year intervals from 1900 to 2000. Figure 4.18 shows the distribution of water users in each group. Compared to the user distribution in the original TCEQ groups, the alternative TCEQ grouping is more evenly distributed. For instance, Group 7 permit holders are divided among three groups in the alternative TCEQ grouping system.

Figure 4.19 shows a comparison of the optimal curtailment hours under the original TCEQ priority grouping and the alternative one. The curtailment hours of Group 1 remain the

same in both grouping systems. There is a slight increase from 0 to 0.5 hours in partial water users and from 0 to 1.5 hours in the rest of Group 2 by implementing the alternative TCEQ priority grouping. Groups 3, 4, and 5 have an incremental increase in the curtailment hours: Group 3 increases from 0 to around 3; Groups 4 and 5 correspondingly increase by approximately 120% and 170%, respectively. The curtailment hours for permit holders in Groups 6 and 7 decrease by approximately 24% and 4%, respectively. These results show that the alternative TCEQ priority grouping is beneficial to the most junior groups, Groups 6 and 7. Allocations to the senior groups such as Groups 1 and 2 are almost independent of the priority grouping system. The alternative TCEQ priority grouping is not favorable to Groups 3, 4, and 5, whose curtailment hours increase in the alternative grouping system. These results demonstrate that the optimal water allocation strategy is highly sensitive to the grouping system and TCEQ decision makers may want to study this approach more closely.

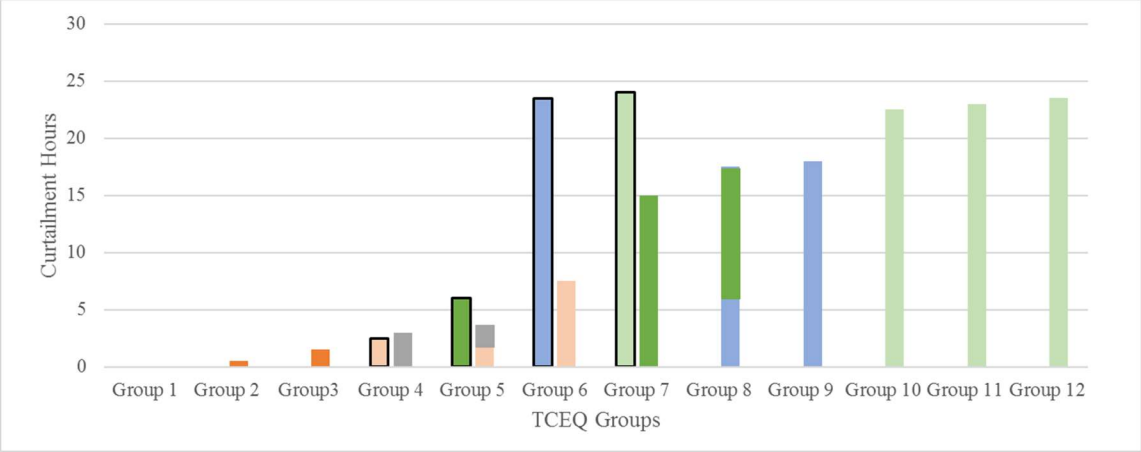

Figure 4.19. Comparison of Optimal Curtailment Hours between the Original and Alternative TCEQ Priority Doctrine Groupings

Impact of Non-Compliance

During water shortages, water users may not obey the optimal water allocation strategies. Non-compliance is defined as the percentage of violations during curtailment hours. Table 4.8 shows a set of non-compliance scenarios, which are defined by various non-compliance ratios for each group of water right permit holders. The violation of minimum river streamflow, which measures the impact of each non-compliance scenario, is also given. We use complete non-compliance (nc-s9 as shown in Table 4.8) as the baseline, which means all users violate the rules. In the first non-compliance scenario, the non-compliance rate decreases from Groups 1 to 7. This

scenario assumes that senior water users are more inclined to violate the optimal water allocation than junior water users since senior water users are eligible for more water. The non-compliance rate is set over the range 100% to 10% from Groups 1 to 7. The violation ratio is around 25% compared to the baseline experiment. This indicates that such non-compliance does not pose serious effects on the river system, because the curtailment hours of senior water users are small and the violation of those hours will not affect the river system seriously.

The non-compliance rate is also in the range of 10% to 100% from Groups 1 to 7 in the second scenario, but in reverse order (i.e., highest non-compliance for junior water users). Compared to the baseline experiment (nc-s9), the violation ratio is around 94%, which almost reaches 100%. This indicates that non-compliance of the junior water users has a greater effect on the river system compared to the senior water users, because the curtailment hours of junior water users in the optimal water allocation strategy are larger. The other scenarios have the same non-compliance rate for all groups of users and the violation ratio increases with the increment of non-compliance rate. Therefore, non-compliance of the junior water users will have a greater effect on the river system compared to the senior water users.

Based on these results, when watermasters inspect the withdrawing behavior of permit holders, we recommend that watermasters focus more on junior water right permit holders since the river system is more severely affected by their violations.

Table 4.8. Effects of Non-Compliance on River System

Non-Compliance Scenarios	Group1	Group2	Group3	Group4	Group5	Group6	Group7	Violation	Ratio
nc-s1	100%	85%	70%	55%	40%	25%	10%	0.59%	26.44%
nc-s2	10%	25%	40%	55%	70%	85%	100%	2.09%	93.67%
nc-s3	10%	10%	10%	10%	10%	10%	10%	0.33%	14.63%
nc-s4	25%	25%	25%	25%	25%	25%	25%	0.78%	35.07%
nc-s5	40%	40%	40%	40%	40%	40%	40%	1.17%	52.31%
nc-s6	55%	55%	55%	55%	55%	55%	55%	1.48%	66.28%
nc-s7	70%	70%	70%	70%	70%	70%	70%	1.76%	78.88%
nc-s8	85%	85%	85%	85%	85%	85%	85%	2.00%	89.48%
nc-s9	100%	100%	100%	100%	100%	100%	100%	2.23%	100.00%
nc-s10	0%	0%	0%	0%	0%	0%	0%	0%	0%

4.4.2 Robust Scenario Analysis

Figure 4.20 shows the robust scenario analysis solution, which represents the case for optimizing across multiple scenarios based on the formulation in Equations (4.8), (4.9), and (4.10). The square dots in Figure 4.20 show the optimal solution for the robust scenario analysis. The box whisker shows the range of curtailment hours for each group of water right holders across each single scenario. The results show that the junior water right permit holders from Groups 3 to 7 are required to completely cut off water usage hours and water users in Group 1 can make full use of the water. The water right permit holders in Group 2 are allowed to use water for five hours on the case study day. Therefore, if TCEQ decision makers have no information about water demand and runoff, they can allocate water based on the robust scenario analysis results to ensure that minimum flows are met.

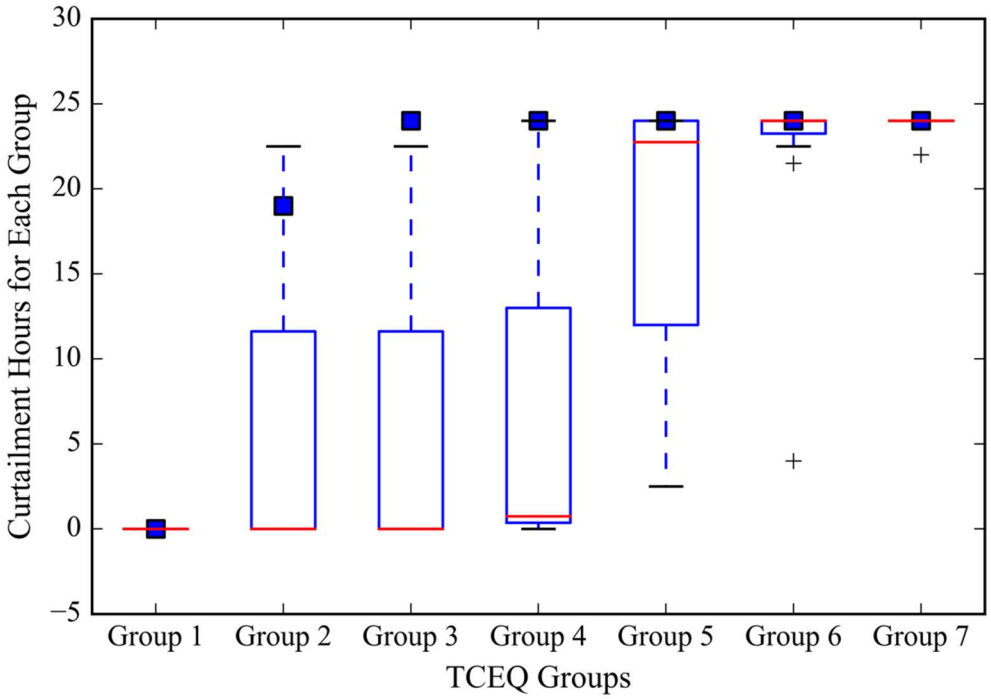

Figure 4.20. Robust Optimization Solution across Multiple Scenarios

Finally, we conduct two more robust scenario experiments with alternative acceptance levels of 80% and 70% in Equation (4.10). Figure 4.21 shows the results of the robust scenario analysis under the three acceptance levels. When the acceptance level is 90%, only water users in Group 1 and a portion of users in Group 2 are allocated water. Under the alternative acceptance

levels, which involve more relaxed constraints, additional water users in Group 2 and a portion of water users in Group 3 are allocated water at acceptance level of 80% and a portion of water users in Groups 2,3, and 4 are allocated water at acceptance level of 70%. When more constraints are relaxed, the violation of the minimum river streamflow increases for each acceptance level from 0.22% to 0.36% to 0.62%. With one more constraint relaxed, the effect on the river streamflow increases by about 70%. Therefore, when the acceptance level decreases, water users in more groups can make use of water but the water shortages lead to significantly higher violations of the minimum river streamflows.

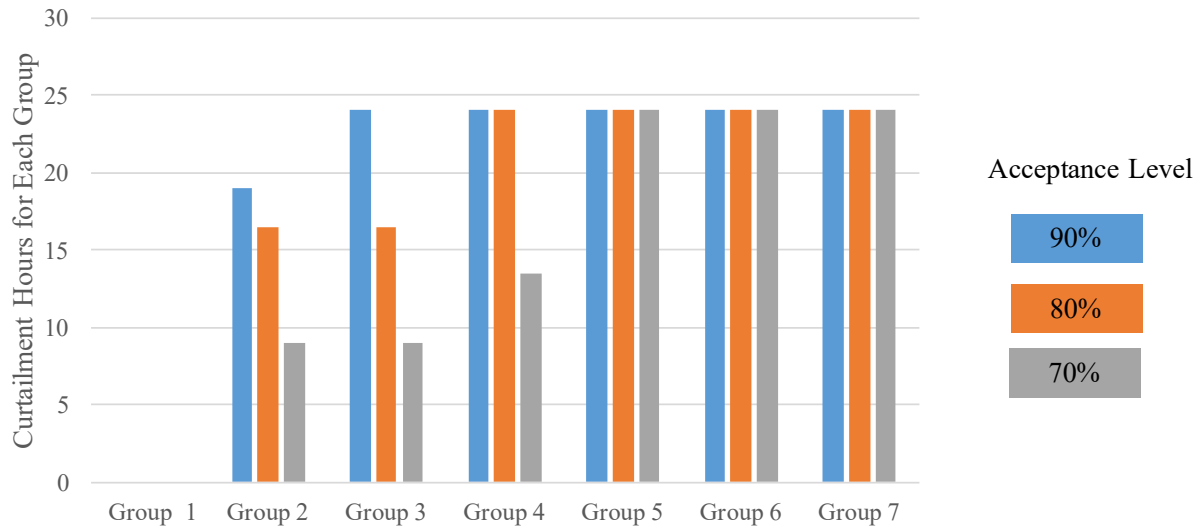

Figure 4.21. Robust Optimization Solutions under Different Acceptance Levels

4.5 Conclusions

This study proposes a service-driven framework of simulation-optimization models for water management during droughts. GA as a metaheuristics optimization approach is used to solve the optimization problem. The simulation and optimization models are published as Web services using NCSA DataWolf and Microsoft Azure ML respectively, then the built model services are coupled to support large-scale water management.

The framework was tested on a water allocation optimization problem in the Upper Guadalupe River Basin, which is aimed at identifying the optimal water allocation strategy under each drought scenario. Single scenario analysis and robust scenario analysis were conducted using the built simulation-optimization model services. The results demonstrate that the most

senior group and the most junior group are insensitive to the uncertainty of water demand and runoff. Both types of uncertainty affect the curtailment hours of other groups. The water right permit holders who are most affected by water demand and weather uncertainty may consider seeking more water from senior permit holders (if a water market were set up) or preserving water in advance to satisfy water demand during droughts. In addition, conservative water demand is suggested for those water users when they call TCEQ to request water.

This work also compares an alternative priority grouping system with the original grouping. The sensitivity of grouping system on the optimal water allocation strategy is high, and the impacts of grouping should be assessed in more detail in future work.

Moreover, non-compliance scenarios are developed and the results have shown that the non-compliance of junior water users is predicted to have a greater effect on the river system compared to non-compliance of senior water users. It is recommended that TCEQ watermasters should pay more attention to junior water right permit holders during inspection.

In addition, a robust scenario analysis that satisfies the constraints under most scenarios was conducted. The optimal solution can be adopted if TCEQ decision makers do not have enough data to accurately estimate water demand and runoff uncertainty.

The current Web application is a prototype, and further development and design of the Web interface is necessary before implementation for decision makers. In addition, the computational effort in handling the constraints, which involve a complex and computationally intensive simulation model, are a major obstacle for real-time Web application. More attention is paid to improving the simulation-optimization services' computational performance in the following chapter.

Chapter 5

Meta-Model Methods for Efficient and Accurate Constrained Nonlinear Optimization

Chapter 4 has developed a Web service-driven framework for water management during droughts (Figure 4.1). However, the computational effort in handling the constraints, which involves executing computationally intensive models, is a major obstacle to developing an effective real-time Web application for decision support. The average execution time for each simulation model evaluation in Section 4.3 (Case Study) is over 30 seconds, thus requiring over 12 hours to obtain the optimal solution. The objective of this chapter is to relieve the computational burden in such constrained non-linear optimization problems by developing a new meta-model approach that enables larger-scale real-time Web services for decision support during severe events such as droughts. We build a framework of meta-model optimization services and implement it using the case study from Chapter 4. The performance of different meta-model approaches is compared with the full simulation-optimization model.

5.1 Introduction

The meta-model approach, which replaces the simulation model with an approximating surrogate function, has been applied to improve computational efficiency in solving complex water resource management problems [e.g., *Yan & Minsker*, 2011, *Pasha & Lansey*, 2010*, Gu et al.,* 2011]. This study develops a new and more efficient meta-model approach using a machine learning classifier to rapidly judge whether a constraint is satisfied. Finally, the meta-model service was built and coupled with the new optimization service for a more efficient real-time Web application (shown in Figure 5.1).

Previous meta-model approaches have focused mainly on replacing the simulation models without considering the model's role in the optimization process. Designing meta-models specifically to efficiently evaluate the constraint of an optimization model is a novel development of this study that supports real-time and interactive Web applications with non-linear optimization services. Water management during droughts requires reliable and immediate information. Take Texas water management as an example. Currently, TCEQ decision makers allocate available water to users based on subjective judgment. To our knowledge, the real-time Web application is the first to implement coupled model services in water resources management

with real-time information retrieval and an optimization approach to improve water allocation strategies.

In this study, both pre-trained and adaptive models are developed. The performance of both models is compared with solutions from the previous chapter for the full simulation optimization model. In addition, the performance of different types of machine learning models is compared using the water allocation optimization problem described in Chapter 4. The built models are published as model services in the Web application given previously, which allows water managers to more efficiently use the water allocation optimization model for decision support.

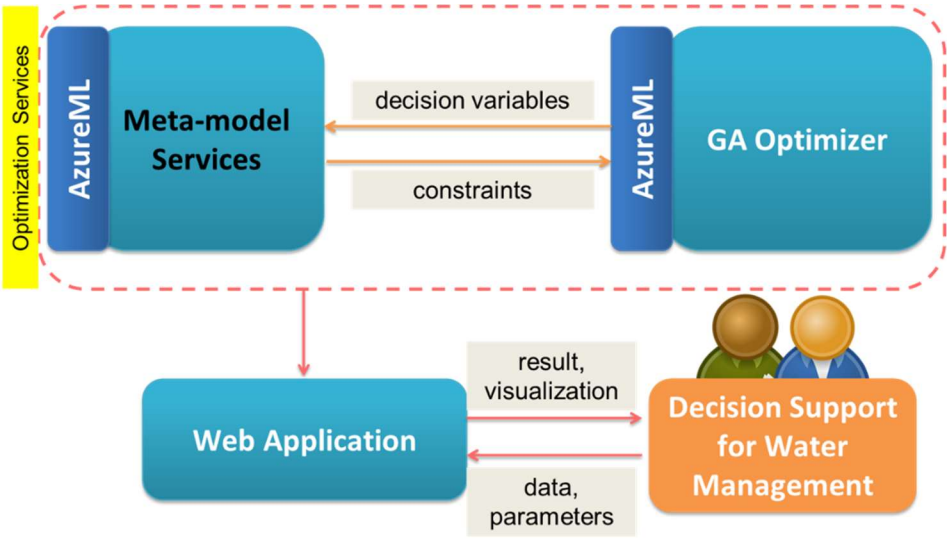

Figure 5.1. Framework for Meta-Model-based Optimization Services During Droughts

5.2 Methodology

Figure 5.2 shows a more detailed diagram of the meta-model genetic algorithm (the dashed box in Figure 5.1). Instead of only using the simulation model to evaluate constraints and calculate fitness values, the meta-model module is adopted for constraint evaluation. Rather than replacing the simulation model with a meta-model that emulates the model's predictions, as in previous work [*Yan & Minsker*, 2011, *Pasha & Lansey*, 2010, *Gu et al.*, 2011], a classifier model is developed that only evaluates whether the constraint is satisfied or not. Two types of meta-model training approaches are considered: pre-trained models [*Johnson & Rogers*, 2000, *Cai et al.*, 2015], which are trained before optimization, and adaptive models [*Yan & Minsker*, 2011, *Wu et al.*, 2015], which are trained and updated during the optimization process.

The "Simulation Model" has been described in Section 4.2.1, River Modeling Service and "GA Operators" is identical to Section 4.2.2, GA Optimizer. The remaining components in Figure 5.2 are discussed in the following subsections.

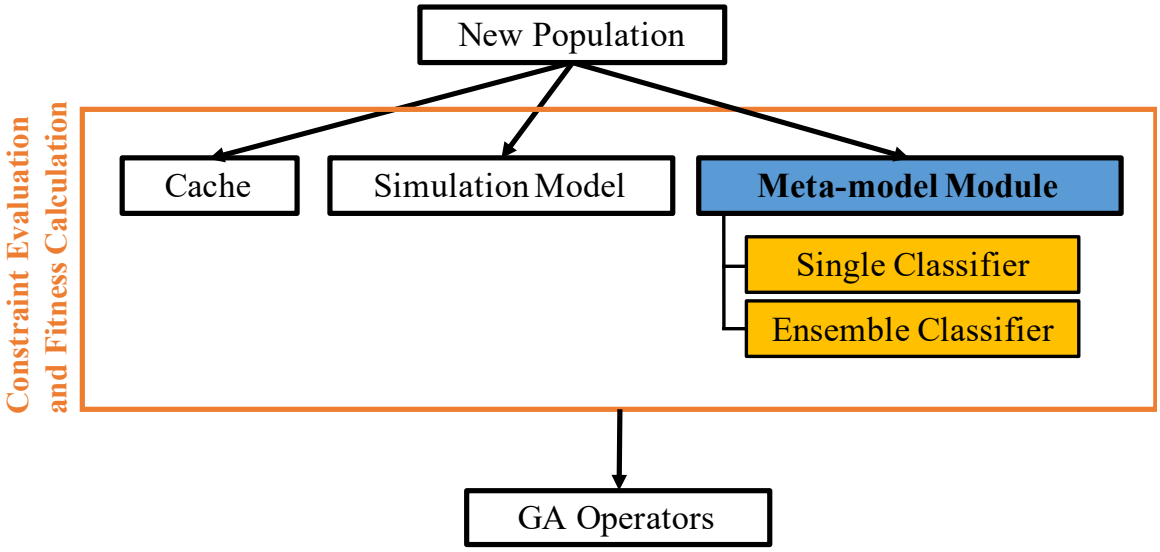

Figure 5.2. Meta-Model Optimization Services

5.2.1 Caching

Caching temporarily stores the simulation-optimization-model chromosomes and the corresponding feasibility of the constraint into a memory space. The evaluated chromosome and the corresponding feasibility of the constraint can be searched and retrieved later. Kratica [1999] has demonstrated that caching improves GA performance by eliminating simulation model evaluations when the evaluated chromosomes reappear. As the population converges in later generations and becomes more homogeneous, caching plays an important role in improving local search accuracy because most evaluations can be based on the cached true values.

5.2.2 Meta-Model Module

The optimization model relies on computationally intensive simulation model evaluations to evaluate the feasibility of the constraints, which determines the fitness of each potential solution. The meta-model module uses machine learning techniques to develop a response function based on training datasets of simulation model runs. The previous approaches use the meta-model to approximate the simulation model function itself. In this work, we explore an alternative approach of building a classifier model to evaluate whether the constraint is satisfied or not using the decision variables as input features. Two types of meta-models, pre-trained meta-model and adaptive meta-model, are described in the following subsections.

Pre-Trained Meta-Model

The pre-trained meta-models are trained offline using sampling data. Two types of data sampling methods are applied to collect training datasets. The first method is to randomly generate datasets and evaluate each random data point using the simulation model. The second method is to generate training datasets from the early generations of one test GA run, which are selected by the early global search operations of the optimization. The classification model is built using the training dataset to evaluate whether the constraint is satisfied or not. Our early tests showed that support vector machine (SVM), neural network (NN), and logistic regression (LR) have advantages in fast training and high accuracy for this application.

Support Vector Machine (SVM)

The SVM algorithm classifies data points by using a maximum margin hyperplane ($\mathbf{w} * \mathbf{x} + b = 0$), which can shift in a perpendicular direction without changing the separation of data points [*Cortes & Vapnik*, 1995]. The SVM decision function depends on support vectors, which are the minimum closest points defining a separating hyperplane (the data points located in the dash line in Figure 5.3). A kernel function, which computes a dot product in high-dimensional feature space, must be selected for SVM [*Ben-Hur & Weston*, 2010]. After some initial experiments, we chose a linear kernel for this study, which is based on a linear discriminant function of the form in Equation (5.1). The sign of the function f($\mathbf{x}$) shown in Equation (5.1) denotes the side of the hyperplane where a point is located.

$$f(\mathbf{x}) = \mathbf{w} * \mathbf{x} + b \tag{5.1}$$

where $\mathbf{w}$ is the weight vector; b is the bias.

The hyperplane in Figure 5.3 separates the feasible and infeasible points. The distance from the infeasible point to the hyperplane (as shown in Figure 5.3) represents the distance to the feasible region, which is used to evaluate the violation of the constraint for each infeasible point in the population during tournament selection.

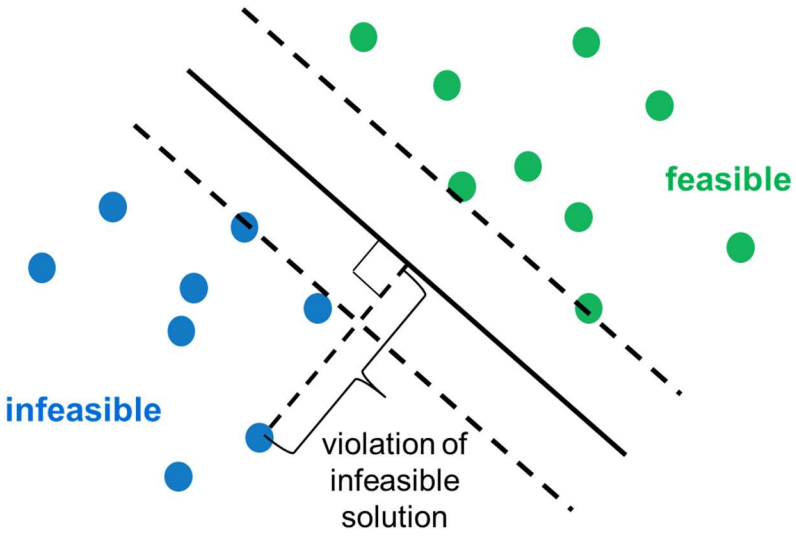

Figure 5.3. Separating Hyperplane with Maximum Margin

Neural Network (NN)

The multilayer perceptron (MLP) is a feedforward neural network that maps the input data onto a set of output labels. Each layer is fully connected to the next one and an activation function is used to map the input of the neuron in one layer to the output of each neuron in the following layer using weight values, which are trained using backpropagation. The MLP network consists of three or more layers (input, output, and hidden layers). The input layer is the input features and the output layer is the output labels (as shown in Figure 5.4), while the hidden layer brings nonlinearity to the approximation using an activation function and weights. The sigmoid function, shown in Equation (5.2), is used as an activation function. The early experiments in this study showed that three layers (i.e., only one hidden layer) were sufficient to achieve good performance. The number of nodes in the hidden layer is tuned during the training process.

$$sig(t) = \frac{1}{1+e^{-t}} \tag{5.2}$$

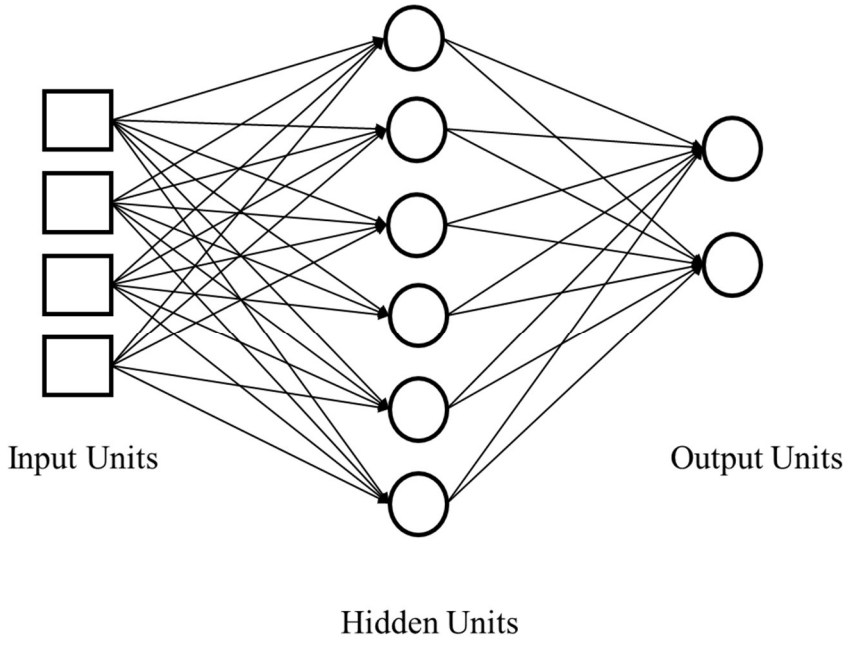

Figure 5.4. MLP Network with Three Layers

Logistic Regression (LR)

The logistic regression (LR) method is used to predict the probability *P(y/**x**)* of a binary response based on input features. A sigmoid function, given in Equation (5.3), is used to calculate the probability of the class label (in this case, feasible or infeasible constraint).

$$P(y = 1|\boldsymbol{x}) = \frac{1}{1+e^{-\boldsymbol{w}^T x}} \tag{5.3}$$

where $\{(\boldsymbol{x}_i, y_i): i = 1, \dots, N,\ y_i \in \{0,1\}\}$ is the training dataset and $\boldsymbol{w}$ is the weighting vector which is learned using maximum likelihood estimation.

The probability of each class label is predicted using Equation (5.1). For example, if $P(y = 1|\boldsymbol{x}) \geq 0.5$, the class label could be set to $y = 1$. However, we found that with the pre-trained meta-model, this approach may identify an infeasible final solution. To solve this issue, a higher probability threshold is used to weigh feasibility over accuracy and ensure that the final solution is feasible. This approach is conservative because the feasible region is narrowed by increasing the probability threshold of feasibility.

Lastly, an ensemble LR model is created that integrates SVM, NN, and LR using majority voting to select the class label. Ensemble models use two or more classifiers to predict the class label. The aggregation of several models can be useful in averaging biases and reducing

prediction variance [*Dietterich*, 2001]. The performance of each single classifier and the ensemble classifier is then compared.

Adaptive Meta-Model

The pre-trained meta-model del may have poor accuracy since the final solution may converge to the infeasible region. To alleviate this difficulty, Yan & Minsker [2011] implemented an adaptive meta-model that adaptively retrains the ANN-based meta-model during optimization. For the groundwater remediation case study examined, they found that adaptive models perform well in saving fitness evaluations while still achieving accurate solutions. This study adopts a similar approach to Yan & Minsker [2011] but focuses on directly predicting constraint feasibility. In addition, this study explores multiple machine learning models to identify which model performs best for the adaptive meta-model GA.

In the initial G generations with the adaptive meta-model, the chromosomes are evaluated using the simulation model. The chromosomes and corresponding feasibility are stored in cache for future use in later generations. A meta-model is then initially trained using this initial training dataset. At each following generation, a sample of chromosomes are evaluated using the simulation model and other chromosomes are evaluated using the meta-model coupled with cache retrieval for chromosomes that have already been evaluated. The sampled simulation solutions are stored in a pool as well as updated in the cache library. When the retraining criteria, the number of training datasets in the pool reaching a pre-defined data size, are satisfied, the meta-model will be retrained. The whole process of the adaptive model approach is shown in Figure 5.5.

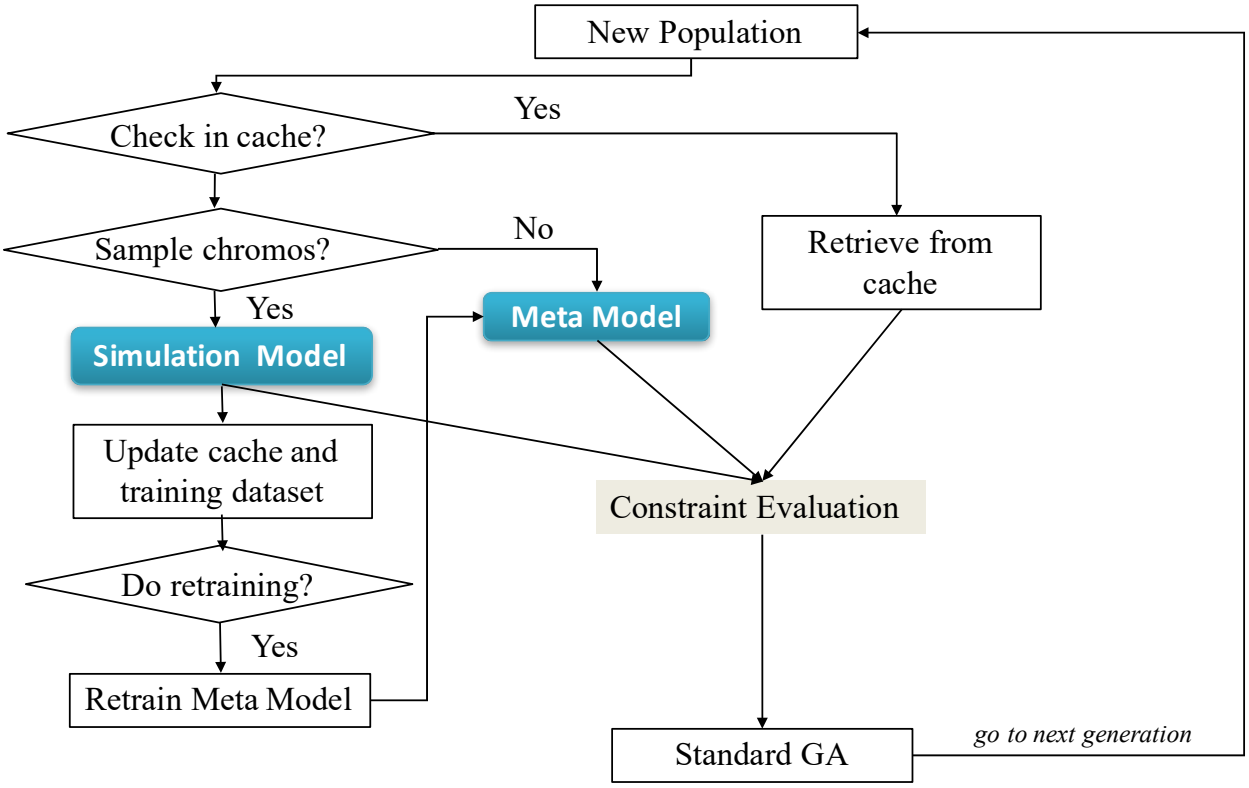

Figure 5.5. Adaptive Meta-Model Framework

5.3 Case Study for Meta-Model Implementation

Scenario 1 with actual runoff, presented in Chapter 4, was used to test the meta-model performance and compare the optimal solution and the computation time for different models. The set-up of the meta-models is discussed in the following subsections.

The real-coded standard GA developed in Chapter 4 was implemented with the same parameters. The population size is 100. The crossover probability is 0.9 and the mutation probability is 0.2. The constraint handling approach is tournament selection method as described in Chapter 4. The meta-models were trained to predict whether Equation (4.7) is satisfied or not. The input features were the curtailment hours for each group of water right holders.

For the SVM model, the distance to the hyperplane was used to determine the violation of the constraint for each infeasible solution. For other algorithms, the probability of a chromosome being infeasible was used to determine which chromosome causes a larger constraint violation. The optimal solution found by the GA for this scenario has total curtailment hours at 56.0.

5.3.1 Pre-Trained Meta-Model

As described in the methodology section, the pre-trained meta-model was built offline before optimization using two types of training datasets. The training data size was set at 400 [*Yan & Minsker*, 2011], considering the model accuracy and computation time of simulation models. The caching was disabled here to examine only the performance of the offline meta-models.

The tuning parameters of each machine learning model were selected based on the predicting accuracy using K-fold cross validation. K=5 was chosen in the study since five- or tenfold cross-validation is recommended as a good compromise [*Breiman & Spector*, 1992, *Kohavi*, 1995]. The one standard error rule of cross validation was adopted to obtain the simplest (most regularized) model, whose error is within one standard error of the minimal error [*Breiman et al.*, 1984].

$$\hat{\theta} = \underset{\theta \in \{\theta_1, \dots \theta_k\}}{\operatorname{argmin}} \; CV(\theta) \quad (5.4)$$

$$CV(\theta) < CV\left(\hat{\theta}\right) + SE\left(\hat{\theta}\right) \quad (5.5)$$

$$SE\left(\hat{\theta}\right) = SD\left(\hat{\theta}\right)/\sqrt{K} \quad (5.6)$$

where θ represents the tuning parameter; $CV(\theta)$ represents the cross-validation error; $SD\left(\hat{\theta}\right)$ represents sample standard deviation of $CV(\theta)$ over K-fold; $SE\left(\hat{\theta}\right)$ represents the standard error of $CV(\theta)$ over K-fold.

In general, we choose the tuning parameter value that minimizes cross-validation errors as shown in Equation (5.4). Instead, we choose the parameter that induces the simplest model while satisfying Equation (5.5).

The machine learning models are implemented using Python scikit-learn library. The tuning parameter for each machine learning algorithm is discussed below.

The tuning parameter C of linear SVM (the equation for C is given in the appendix), which controls the range of the margin (the distance between the dashed lines in Figure 5.3), was tested among 0.1, 1, and 10. C is a regularization parameter that controls the trade-off between a small error in the training dataset and the generalization of the classifier [*Hsu et al.,* 2003]. A smaller value will lead to a larger margin that may misclassify data points, and a larger value will lead to a smaller-margin hyperplane, which will classify all training points correctly. C =0.1 was selected for both training datasets based on the results of cross validation.

The NN consists of three layers: one input layer, one hidden layer and one output layer. We tested classifier accuracy with different values of the NN tuning parameter from 5 to 15. The best number of nodes for Data Type 1, which is generated using the first sampling method described in Section 5.2.2 (pre-trained meta-model) is 5 and the best number of nodes for Data Type 2 is 7.

The tuning parameter for logistic regression that controls the regularization was selected as 1×10^{-5}.

5.3.2 Adaptive Meta-Model

The adaptive meta-model GA integrates simulation model evaluation, meta-model evaluation, and caching into an integrated framework. There are three parameters for the adaptive model: the initial training generations, the sampling ratio, and the retraining data size. The tuning of each parameter is explained below.

The number of initial training generations determines the initial training data size because the initial meta-model is built based on the training data sets of the initial generations. Yan & Minsker [2011] demonstrated that AMGA's performance is insensitive to the size of the initial training set and a coarse fitness estimation at an early stage is sufficient for the GA to detect promising regions of the solution space. Therefore, a smaller value of the number of initial generations is sufficiently accurate and leads to fewer simulation model evaluations in the initial stage. This study used the chromosomes from the first three generations as the initial training dataset as suggested by Yan & Minsker [2011].

The sampling ratio determines the overall sampling level of the adaptive model and the retraining frequency of the meta-models. A low sampling ratio leads to longer retraining intervals and the coarse prediction of the initial meta-model will be retrained for more generations. A high sampling ratio will lead to longer computation time due to more simulation model evaluations. After trial and error experiments, the sampling ratio of 0.1, which means 10% of the chromosomes in each generation will be evaluated using the simulation model, was chosen for this study. A random sampling strategy [*Yan & Minsker*, 2011] was used to sample chromosomes where any chromosome in the population has an equal likelihood of being sampled.

The retraining data size was set to 150. When the retraining pool is full (i.e., reaches the user-specified retraining data size), the meta-model will be retrained and updated. Then the retraining pool is emptied for the next retraining.

5.4 Results and Discussion

This session presents the results for the pre-trained and adaptive meta-model tests. Each model was implemented for three random seeds and the model with the best performance among them was selected and discussed in this section.

5.4.1 Pre-Trained Meta-Model

The pre-trained meta-models are trained using two types of datasets: the randomly generated datasets (named Data Type 1) and the datasets from the early generations of one test GA run (named Data Type 2). Tables 5.1 and 5.2 show the optimal solutions of each type of classifier for each dataset respectively.

Table 5.1 shows the optimal solutions of each classifier trained using the randomly generated dataset (Data Type 1). For the SVM classifier, if the prob_threshold is set at 0.5 (called the "regular model"), it will converge to an infeasible solution. However, if the prob_threshold increases to 0.8 (called the "conservative model"), it will converge to a feasible solution.

Figure 5.6 shows the simulation model evaluation of the best solution for each generation with pre-trained meta-model optimization. Figure 5.6.a shows that after twelve generations, the best solution chosen by the regular SVM model has moved to an infeasible region and never returns to the feasible region. Figure 5.6.b shows the results for the conservative SVM model. Although the best solution for a few generations falls into the infeasible region (the region below the red line in Figure 5.6.b), the higher probability threshold brings the best solution back to the feasible region (above the red line in Figure 5.6.b). The NN classifier has a similar trend as SVM. For other classifier models, both the regular and conservative models can achieve feasible solutions.

Table 5.1. Optimal Solutions of Pre-Trained Meta-Model for Data Type 1

	Probability threshold	fitness	Group 1	Group 2	Group 3	Group 4	Group 5	Group 6	Group 7	feasibility
SVM	0.5	56.5	0	0	0	0.5	11	21	24	N
	0.8	58.0	0	0	0	0	14.5	19.5	24	Y

Table 5.1 (cont.)

	Probability threshold	fitness	Group 1	Group 2	Group 3	Group 4	Group 5	Group 6	Group 7	feasibility
NN	0.5	56.5	0	0	0	0.5	11.5	20.5	24	N
	0.8	58.0	0	0	0	1	13	20	24	Y
LR	0.5	58.5	0	0	0	2	10	22.5	24	Y
	0.8	59.0	0	0	0	2	11	22	24	Y
Ensemble	0.5	58.0	0	0	0	1	13	20	24	Y
	0.8	59.0	0	0.5	0.5	1	11	22	24	Y

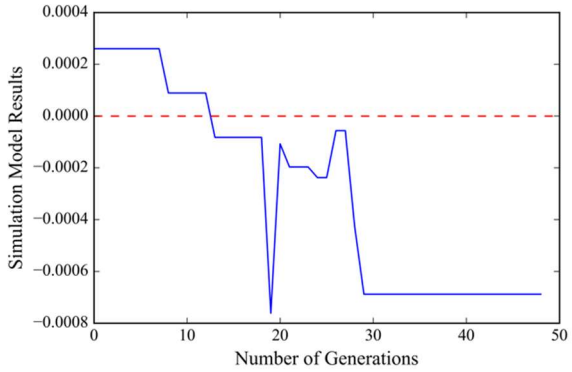

Figure 5.6.a. Simulation Result for SVM with Prob_threshold = 0.5

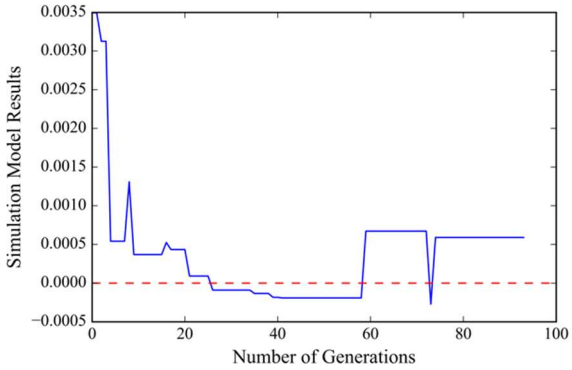

Figure 5.6.b. Simulation Result for SVM with Prob_threshold = 0.8

Figure 5.6. Simulation Result for Best Chromosome of Each Generation for Pre-trained SVM Model

Table 5.2 shows the optimal solution of each classifier for Data Type 2, which has a similar trend to Data Type 1. The conservative SVM and NN classifier models perform better than the regular ones in identifying feasible solutions. For other classifiers, both probability thresholds can achieve feasible solutions. The conservative LR model and the conservative ensemble model perform better than the corresponding regular model. Therefore, the conservative models are recommended for SVM and NN classifiers to enforce feasibility.

Table 5.2. Optimal Solutions of Pre-Trained Meta-Model for Data Type 2

	Probability threshold	fitness	Group 1	Group 2	Group 3	Group 4	Group 5	Group 6	Group 7	feasibility
SVM	0.5	57.5	0	0	0	0.5	13.5	20.5	23	N
	0.8	58.0	0	0	0	0.5	13	21	23.5	Y
NN	0.5	57.0	0	0	0	0	13	20	24	N
	0.8	58.0	0	0.5	0.5	1.5	9.5	22	24	Y
LR	0.5	62.0	0	0	0	3.5	16.5	19.5	22.5	Y
	0.8	58.5	0	0	0.5	0.5	14.5	19	24	Y
Ensemble	0.5	58.5	0	0	0.5	0.5	14.5	19	24	Y
	0.8	57.5	0	0	0	1.5	10	22	24	Y

Figure 5.7 summarizes the results of each classifier for each type of training dataset over three random seeds. For randomly generated datasets (Data Type 1), the best classifiers in terms of the fitness value are the conservative SVM, NN model and the regular ensemble model. For datasets generated by the initial optimization (Data Type 2), the best classifier in terms of fitness is the conservative ensemble model. Again, the conservative model is helpful when the original model cannot find the feasible solution. By weighting feasibility more heavily than accuracy, the conservative optimization can help direct the search from the infeasible region back to the feasible region.

In addition, the ensemble classifier model performs better than the single classifier for both Data Types 1 and 2. Comparing the two types of data, the datasets originating from the initial optimization perform better than randomly generated datasets. During the initial generations of GA, the individuals with higher fitness are statistically selected more often to be parents and generate the next generation using a GA operator. The average fitness of the datasets from the initial generations of GA (Data Type 2) is higher than that of the randomly generated datasets (Data Type 1), which leads to better performance for Data Type 2.

Moreover, compared to the simulation-optimization model results in Chapter 4, the optimal solutions with the pre-trained meta-model have not converged to the optimal solution (fitness value = 56.0). It only converges to a near-optimal solution (best fitness value = 57.5 among all classifiers). However, the computation time of the pre-trained meta-model has an overwhelming advantage. The execution time for running the simulation models to obtain the training dataset is around three hours, while evaluating the classification model requires just 100 seconds. Therefore, the comparison of computation time is given in terms of the number of simulation model runs, which dominate the computation time. The computational efficiency of the pre-trained meta-model GA exceeds the simulation-optimization model.

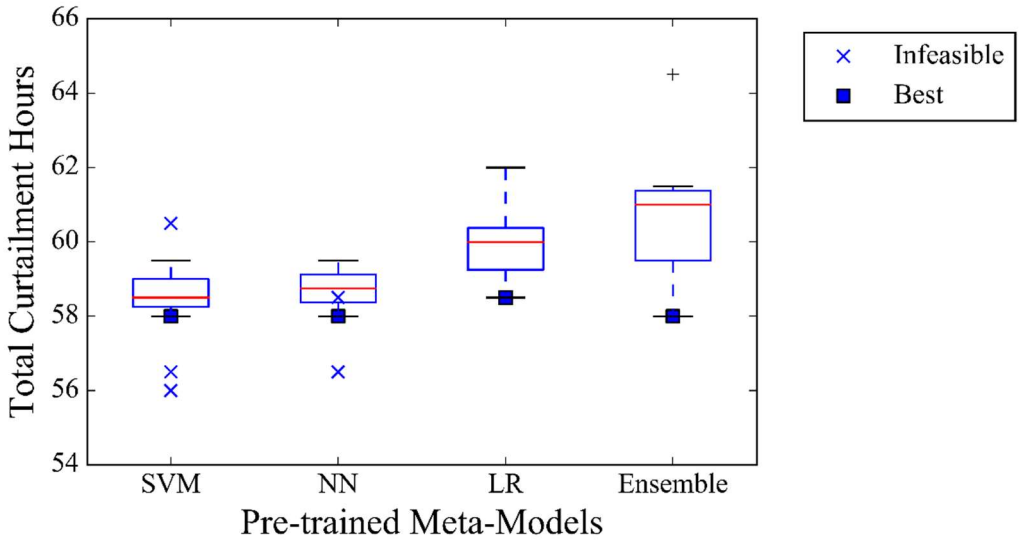

Figure 5.7.a. Performance of Pre-Trained Meta-Model for Data Type 1

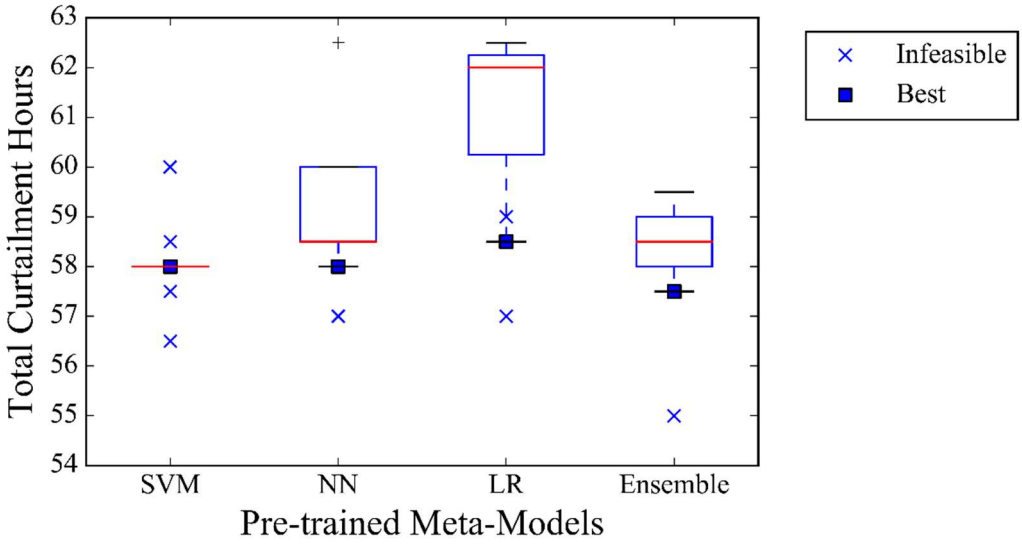

Figure 5.7.b. Performance of Pre-Trained Meta-Model for Data Type 2
Figure 5.7. Performance of Pre-Trained Meta-Model

5.4.2 Adaptive Model

The results of the adaptive meta-model approach for each classifier are shown in Table 5.3. The adaptive meta-model GA consistently converges to the optimal solution found with the simulation optimization model in Chapter 4. Since all single classifiers converged to the optimal solution, the ensemble classifier and the conservative constraint approach are not necessary.

Figure 5.8 shows the number of evaluations for each evaluation approach. In the adaptive meta-model GA, caching is enabled. For the initial generations, the simulation model dominates the evaluations. Then the meta-model becomes active and finally, as more evaluations

are accumulated in the cache and convergence limits the search, caching becomes more active in later generations. Since cached evaluations in later generations are from simulation model evaluations, the optimal solutions from the adaptive meta-model GA are more accurate than the pre-trained meta-models and the optimal solution found by the adaptive model is not sensitive to the choice of classifier. All three different classifiers give the same fitness value for the same random seed, indicating that the adaptive model is not affected by the choice of classification model.

In terms of computation time, the number of simulation model evaluations is around 740, and the adaptive meta-model GA takes eight hours to converge to the optimal solution. Overall, the adaptive meta-model GA performs better than the simulation model GA in computational efficiency while maintaining accuracy. Compared with pre-trained meta-models, the adaptive meta-model GA is better in model accuracy but worse in computation time.

Table 5.3. Optimal Solution using Adaptive Meta-Model Optimization

	Fitness	Group 1	Group 2	Group 3	Group 4	Group 5	Group 6	Group 7
SVM	56	0	0	0	0.5	9.5	22	24
NN	56	0	0	0	0.5	9.5	22	24
LR	56	0	0	0.5	1	7	23.5	24

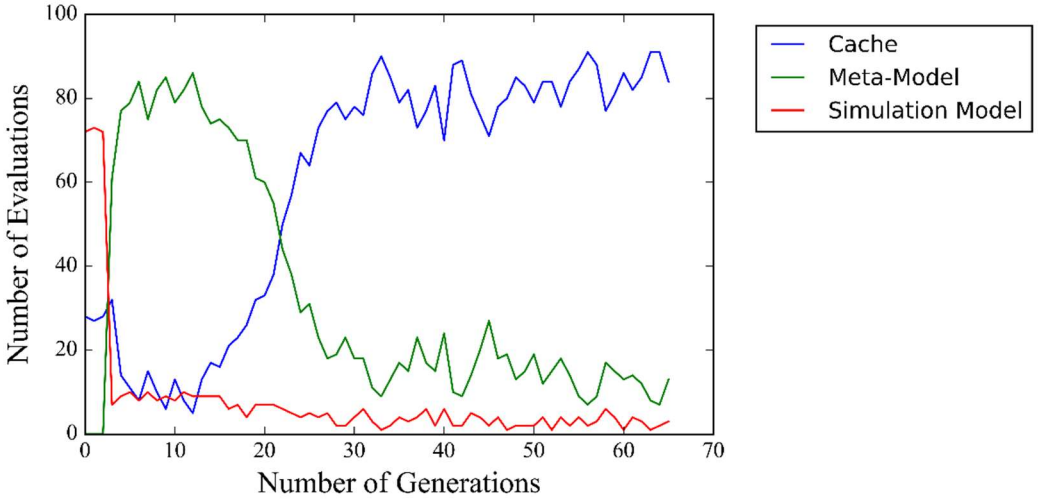

Figure 5.8. Progression of Each Evaluation Approach in Adaptive Neural Network-based Meta-Model Optimization

5.4.3 Overall Comparison of Best Meta-Model Approaches

Figure 5.9 shows a comparison of the simulation model GA, the pre-trained meta-model GA, and the adaptive meta-model GA, using the number of simulation model evaluations as a metric for computation time. Since each simulation model run takes approximately 30 seconds,

the time for simulation model evaluations dominates computational time and the time for other steps can be neglected here. The converged fitness values of the simulation model GA and the adaptive meta-model GA are the same and the pre-trained meta-model GA performs worse, but only by 2.6%. The adaptive meta-model GA can save around 58% of computation time to solve the same problem while still achieving the same optimal solutions. The pre-trained meta-model GA reduces computation time by 78% but does not reliably obtain the optimal solution. Since the adaptive meta-model approach stores the dataset for retraining the meta-model with a small percentage of chromosomes over each generation, this leads to a larger number of simulation model evaluations but improved accuracy, as shown in Figure 5.9.

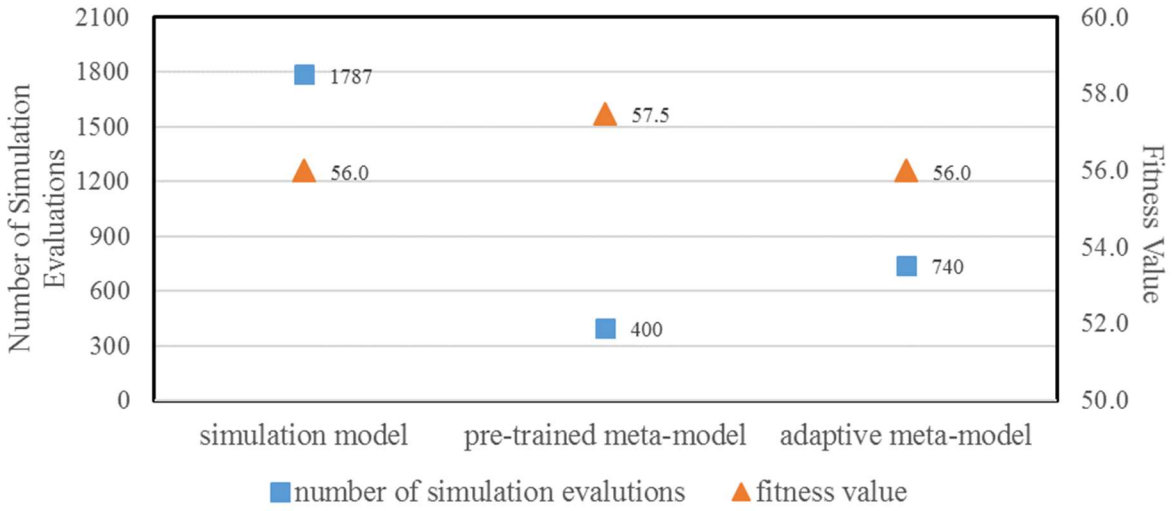

Figure 5.9. Comparison of Three Models

Figure 5.10 compares the performance of each type of model if the number of simulation model evaluations is the same. When the number of simulation model evaluations in the pre-trained meta-model GA increases to 740 (same as the adaptive meta-model GA), the optimal solution found by the pre-trained meta-model GA is improved from 57.5 to 56.5 hours due to the larger training data size. The performance of the simulation model GA is significantly worse, increasing to 63.0 hours, since the number of evaluations is not sufficient to achieve convergence. Therefore, the overall performance of the adaptive meta-model GA is the best-performing algorithm for the same number of simulation evaluations.

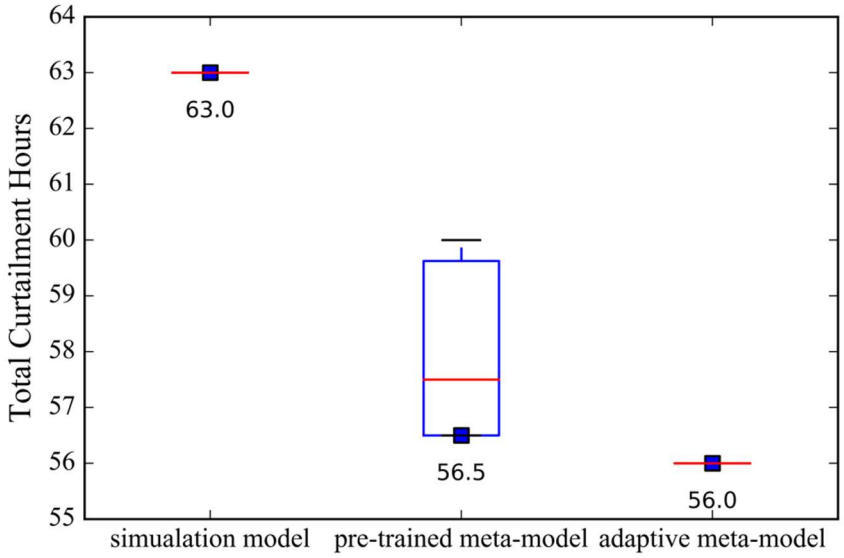

Figure 5.10. Comparison of Fitness Value with Same Number of Simulation Evaluations

In addition, we also compare the meta-model developed in this work with the previous approach that estimates the simulation model predictions [*Yan & Minsker*, 2011, *Pasha & Lansey*, 2010, *Gu et al*., 2011]. Instead of the classification-based constraint approach, a regression-based adaptive approach using a neural network model is built to predict the value of the left side of Equation (4.7). The same approach as Yan & Minsker [2011] is applied using the framework shown in Figure 5.5.and described in Section 5.2.2. The first three generations are used as the training dataset to build the initial regression model, following the approach of *Yan & Minsker* [2011]. In each generation, a sample of chromosomes are selected for evaluation with the simulation model using their random sampling strategy, where any chromosome in the population has an equal likelihood of being sampled. In this work, the sampling ratio is equal to 0.1 as suggested by Yan & Minsker [2011], meaning that 10 percent of the chromosomes are assessed with the simulation model. The sampled simulation solutions are stored in a retraining pool. The regression model is retrained and updated when the retraining pool reaches the retraining data size, which is 150 [*Yan & Minsker,* 2011].

The results of applying this approach for this case study are identical to the results given previously, thus validating that the approach taken in this work is correct. However, the computational time of the regression-based approach is 12% faster than the classification-based

approach. The regression-based approach takes fewer generations to reach the optimal solution than the classification-based approach for this case study. The number of generations to converge is 56 for the regression-based approach and 65 for the classification-based approach with the same population size of 150). Future research is needed to further test the new proposed meta-model approach for other simulation-optimization problems and compare the performance and computation time of both approaches.

The Web application shown in the previous chapter is modified to provide meta-model services coupled with the optimization model services, using the adaptive meta-model approach. The pre-trained meta-model approach requires users to collect the training dataset, executes the simulation model and builds the meta-model offline, which requires technical expertise from users that cannot be automated and generalized in a Web application. Therefore, the adaptive meta-model approach is implemented in the Web application as it performs best in terms of trade-offs between model accuracy and computation time, and the application is easily generalized to other case studies.

Figure 5.11 shows the prototype Web application for real-time water allocation decision support services. The uploaded "river basin information" is an input file to the river modeling service. The uploaded "water demand information" file provides the user information as shown in Figure 5.11.a and the corresponding diversion values over the river system. Therefore, the Web application can be easily extended to other river basins by changing these files.

The "Water Request Date" is an input parameter used for the river modeling system to download input data (runoff) from NWS. Since the water allocation strategy is sensitive to the permit grouping as discussed in Section 4.4.1 (Alternative TCEQ Priority Grouping), the Web application allows users to determine the number of permit groups by providing the priority start date and end date as well as setting the number of groups. The model services can be executed on a remote server in the Cloud by uploading the input files, filling in the input parameters, and selecting the "Submit" button; the result (the optimal curtailment hours for each group of water right permit holders) is retrieved and shown in the Web interface "Water Allocation Recommendation" as shown in Figure 5.11.b. The Web application provides an easy way for decision makers to rapidly allocate water and explore multiple permit grouping systems.

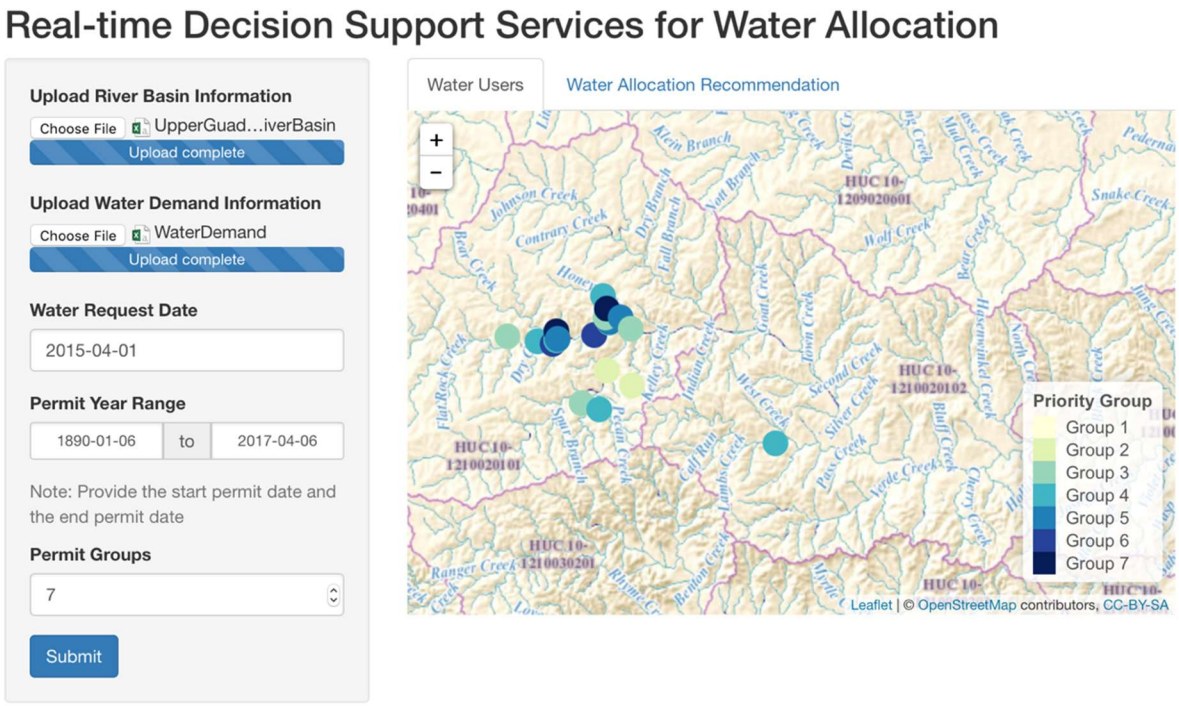

Figure 5.11.a. Web Interface for Water User Information

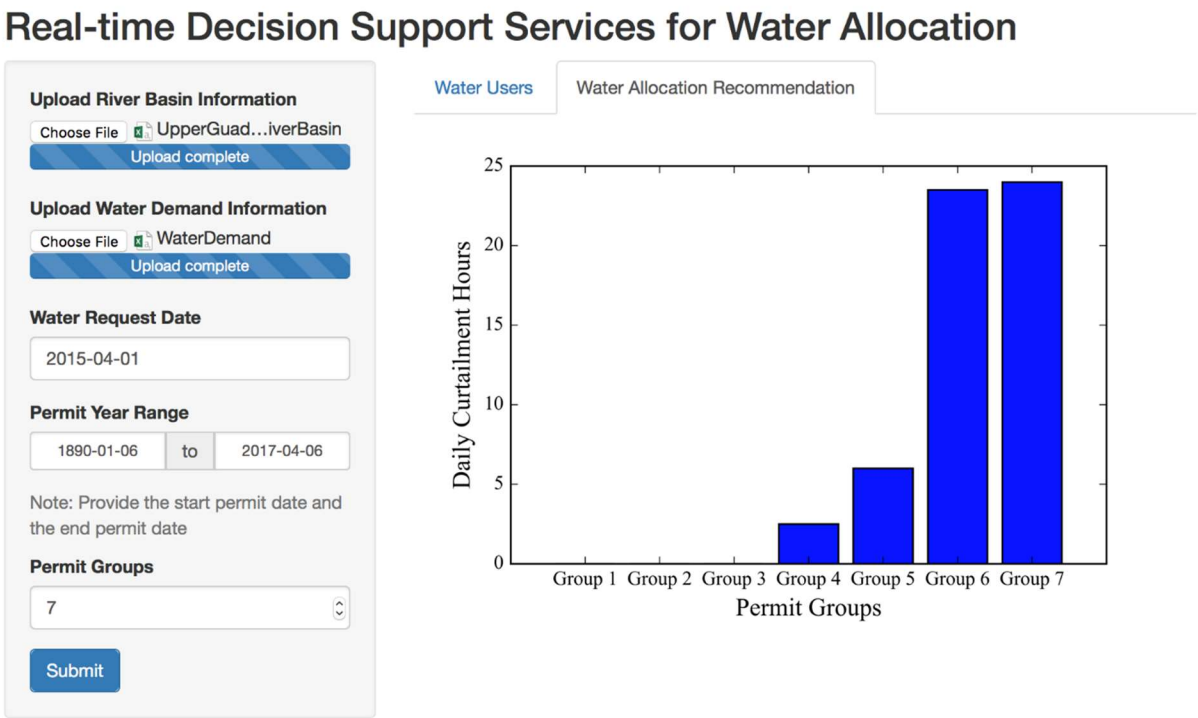

Figure 5.11.b. Web Interface for Water Allocation Recommendations

Figure 5.11. Web Interface for Real-time Water Allocation Decision Support Services

5.5 Conclusions

In this study, we investigated a new meta-model GA approach to replace the simulation model in optimization problems, and compared pre-trained and adaptive versions in terms of computation time and searching accuracy. Instead of building the meta-model to replace the simulation model, the meta-model approach in this study considers the model's role in the optimization and uses a classification model to evaluate the simulation model-based constraint in the optimization formulation. The developed meta-model framework was implemented in water allocation optimization problems in the Upper Guadalupe River Basin.

We compared four types of classifiers: SVM, NN, LR, and the ensemble of each classifier. A more conservative optimization approach, which weights feasibility more heavily than accuracy by increasing the probability threshold from 50% to 80%, was needed in some cases to ensure convergence to the optimal solution. The results demonstrate that the conservative approach performs better for ensuring constraint satisfaction in the SVM and NN classifiers. However, if both approaches converge to a feasible solution, the conservative approach does not consistently perform better.

The pre-trained models were trained based on two types of datasets: randomly generated datasets and datasets generated by an initial optimization run. The datasets from the initial generations of a GA perform better as the average fitness value is higher than the randomly generated datasets. Overall, the results demonstrate that the pre-trained meta-model converges to the near-optimal solution with less computation time. It is suggested that the conservative option as well as the choice of classifier should be further tested to obtain the best performing meta-model.

The adaptive meta-model GA integrates the simulation model, the meta-model, and caching into an efficient framework. The simulation model dominates in early generations and then the meta-model is active in later generations. Caching becomes more active and dominant in later generations, which ensures that the adaptive meta-model GA converges to the optimal solution while saving computation time. The adaptive meta-model GA is insensitive to the choice of the classifier, as different types of single classifiers give the same optimal solution. The adaptive meta-model GA performs best considering the trade-offs between computation time and searching accuracy, as compared to the pre-trained meta-model and simulation model GA.

A prototype Web application is developed that couples meta-model services with optimization model services. The adaptive meta-model is implemented in the Web application as it does not require preliminary work such as selecting classifiers, tuning parameters, and executing simulation models offline. The Web application allows decision makers to explore optimal water allocation strategies using different permit grouping systems, and can be easily extended to other regions by uploading the river system and water demand information.This work extends the approach of Yan & Minsker [2011] by implementing a classification-model-based meta-model approach for water allocation optimization problems. The results demonstrate that a meta-model approach is promising in reducing computation time for water allocation simulation-optimization models. This approach can be generalized to other models by updating the training datasets. In addition, the service-driven approach allows easy integration with other models to solve water resource problems. Lastly, the current Web application is a prototype and further development is necessary before the system can be adopted for water allocation management.

Chapter 6

Conclusions and Future Work

The previous chapters discuss the application of a service-driven approach to decision support in water management, using case studies related to drought and flooding. This chapter summarizes the findings as well as introduces some future topics for extending this work.

6.1 Conclusions

When extreme events happen, rapid response to assist in water management requires reliable and immediate information collection, optimal model-based water operations, and other information. Disparate models such as meteorological models, hydrological simulation models, water operations models, and other models are needed in this process. But the configuration of each model and communication of different models has been a major obstacle in applying them to this process. This dissertation develops a service-driven approach, which deploys each model in a loosely-coupled environment where each model service is an individual component and information is exchanged among each component across the whole network. The modeling components, which require complex configuration and specific data standards, can be integrated into a decision support system with each model located in its own running environment and easily accessible through Web services. We developed a data-driven model service to rapidly estimate reservoir inflows for flooding events, and a simulation-optimization model service to optimize water allocation under multiple drought scenarios. To relieve the computational burden of the simulation-based optimization service without sacrificing accuracy, a meta-model service coupled with an optimization model service was proposed to solve non-linear constrained optimization problems.

The data-driven model service was applied to flooding events in the Lower Colorado River Basin in Texas in November 2014 and May 2015, which involved a sudden switch from drought to flooding. Two prediction models were constructed: a statistical model for flow prediction and a hybrid statistical- and physics-based model that estimates errors in flow predictions from a physics-based model. The results demonstrate that the statistical flow prediction model provides acceptably accurate short-term forecasts. However, for longer-term prediction (2 hours or more), the hybrid model forecasts more accurately than the purely

statistical or physics-based prediction models alone. The Web services based on these built models use Microsoft's Azure Machine Learning software and are accessible through a browser-based Web application, enabling ease of use by both technical and non-technical personnel.

The second section of the thesis focuses on water allocation problems during droughts. The imbalance between water supply and water demand poses a crucial challenge for water management. The coupled simulation-optimization models have been used widely in water resource management, but the service-driven approach for simulation-optimization was first proposed. This approach allows an optimization service to communicate with a simulation service using standards-based approaches that are independent of the model structure. The resulting framework can be published as a Web application to enable near real-time decision support in the Cloud. A drought event in the Upper Guadalupe River Basin in April 2015 was used as a case study to illustrate the benefits of the approach. The Texas Commission on Environmental Quality (TCEQ) currently allocates water in the basin based on subjective judgment and does not have quantitative methods for rapid daily water allocation based on the best available forecasts of river streamflow. The current TCEQ grouping system divides water right permit holders into seven groups to allocate water in a manageable way. The built simulation-optimization model services were implemented to identify the optimal water allocation under each weather and water demand uncertainty scenario.

The results demonstrate that the most senior group and the most junior group are insensitive to the uncertainty of water demand and runoff. Both types of uncertainty affect the curtailment hours of other groups. The water permit holders who are mainly affected by water demand uncertainty under each drought situation may consider seeking more water from senior water right permit holders or preserving water in advance to satisfy water demand during droughts. Moreover, conservation measures to reduce water demand are suggested to aid those water users.

The results also provide suggestions to decision makers for more effective water allocation management. The sensitivity of the permit grouping system on optimal water allocation strategies suggests that TCEQ water masters should pay more attention to the grouping system. Since the non-compliance of junior water right permit holders has a greater effect on the river system, it is recommended that TCEQ water masters focus on junior water permit holders during inspection. In addition, the optimal solution from robust scenario analysis

solution would provide a more robust strategy if TCEQ decision makers have little information about water demand and runoff uncertainty. Moreover, a prototype Web application was developed for near real-time implementation in water allocation management.

Results from the application of simulation-based optimization model services indicate that the computational effort in handling the constraints, which involve a complex, computationally intensive simulation model, are a major obstacle for real-time Web application. Hence relieving the computational burden in constrained non-linear optimization problems is another research objective in this dissertation. Two types of meta-models were developed: a pre-trained meta-model that was built before the optimization and an adaptive meta-model that was built and updated during the optimization process. The meta-model is developed as a classifer that evaluates whether a constraint is satisfied or not in the optimization formulation, which better considers the meta model's role in water management application.

The conservative model, which narrows the feasible region by increasing the constraint probability threshold, was needed with the pre-trained meta-model to aid in converging to a feasible near-optimal solution, but not for the adaptive meta model. The meta-model framework was tested in the same case study as the simulation-optimization model services in Chapter 4 and the performance of different approaches was compared. The results show that the adaptive meta-model GA performs best in model accuracy in addition to reducing computation time by 58%. It also does not require users to select classifiers, tune parameters, or execute simulation models offline. Therefore, a prototype Web interface was designed based on the adaptive meta-model approach to assist decision makers with real-time water management. The Web application allows decision makers to explore optimal water allocation strategies using different permit grouping systems by providing "the permit start date", "the permit end date", and "the number of permit groups". The model services can be easily extended to other regions by updating the river system information and providing water demand data.

6.2 Future Work

The results of this research have demonstrated the promise of a service-driven approach to water management as well as a new meta-model approach to improve water resource optimization. A few future topics are recommended to extend the work.

Firstly, the Web application in this study is a prototype and further user-centered design and development is necessary before the application is adopted for real-world water management. Feedback from LCRA reservoir operators and TCEQ water managers can be used to improve the interface and add more features to allow for real-time decision support. In addition, the parallel execution of model and optimization services has not been explored in the Web application. The future work can be extended to parallel execute model services to improve computational efficiency.

Secondly, in terms of the data-driven model services for predicting reservoir inflows, the built models can be readily updated and improved using the AzureML framework. The prediction data of soil moisture, precipitation, and upstream reservoir *flowout* may be considered to replace historical data to improve reservoir inflow forecast. Moreover, the simulation-based optimization model services were developed using hypothetical water demand data. If real water demand data is available in the future, the framework can be extended to replace the hypothetical data with real data and verify the optimization results using the TCEQ subjective water allocation decisions.

Thirdly, the built framework can be extended to other areas. The water allocation based on Priority Doctrine has been implemented in different U.S. states. The built framework can be modified for other water allocation problems in other areas with a similar Priority Doctrine. The RAPID model service has been applied to the whole U.S. area. By providing the water demand data and corresponding user information, the models and tools developed in this work can be generalized to other regions.

Fourthly, Genetic Algorithm (GA) as a population-based algorithm has been widely applied in water resources management [*Maier et al.*, 2014, *Nicklow et al*, 2010]. Other types of metaheuristic algorithms can be implemented to compare the performance of different algorithms. For instance, the trajectory-based metaheuristic algorithm, which starts from a guess solution and moves to the next solution based on whether performance is improved or not, can be applied to compare the algorithm performance in terms of model accuracy and computation time.

Finally, the optimization model developed in this study only considers the TCEQ Priority Doctrine and the ecological influence on the river system. A single-objective optimization model was implemented to obtain a single optimal solution. The future work can be extended to build multi-objective optimization problems that incorporate more objectives, such as maximizing the

total economic value of water usage. Then, a set of compromise solutions instead of a single optimal solution can be obtained for water resource management. For instance, Rezapour et al. [2013] developed a multi-objective model to maximize the reliability and minimize the water supply cost in water allocation problems. The optimal trade-off between multiple objectives was presented in their work. The service-driven approach for multi-objective optimization should incorporate the decision makers' perspective in the Web application to balance the trade-off among multiple objectives.

APPENDIX

Applied Machine Learning Algorithms

BRT Algorithm (Friedman 2001, Hastie & Friedman, 2008)

Input: training dataset $\{(x_i, y_i)\}_{i=1}^{n}$ where x_i represents input datasets ('features') and y_i represents output dataset ('targets'), number of iterations.

Algorithm:

1. Initialize model with a constant value

$$F_0(x) = argmin \sum_{i=1}^{n} L(y_i, F(x_i))$$

2. For each iteration:
    a. Compute pseudo-residuals:

$$r_{im} = -\left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}\right] \text{ for i = 1,…, n}$$

    b. Fit a decision tree learner $f_t(x)$ to pseudo-residuals using the training dataset.
    c. Add $f_t(x)$ to the model $F_t(x) = F_{t-1}(x) + \epsilon f_t(x)$, where ϵ is called step-size or shrinkage. In this study, it was set to 0.1 to prevent overfitting by not doing a full optimization in each step.
3. Output $F_t(x)$

Linear SVM Algorithm with soft margin (Cortes & Vapnik, 1995)

Input: training dataset $\{(x_i, y_i)\}_{i=1}^{n}$ where x_i represents input datasets ('features') and y_i represents output dataset ('targets'), and $y_i \in \{-1,1\}$, each indicating the class to which the point x_i belong.

The training dataset is used to learn a classifier

$$\mathrm{f}(\mathbf{x}) = \mathbf{w} * \mathbf{x} + \mathrm{b} \begin{cases} \geq 0 & y_i = 1 \\ \leq 0 & y_i = -1 \end{cases}$$

where **w** is the weight vector; b is the bias.

Instead of finding a single hyperplane line to classify the data points, we would prefer a larger margin for generalization. We want to find the "maximum-margin hyperplane" that separate the group of points x_i for $y_i = 1$ from the group of points for $y_i = -1$. The maximum margin solution will be most stable under perturbations of the inputs.

Learning an SVM can be formulated to be an unconstrained optimization problem over **w.**

$$\min ||\mathbf{w}||^2 + C \sum_{i=1}^{n} \max(0, 1 - y_i f(x_i))$$

where $||\mathbf{w}||^2$ controls the regularization; $\max(0, 1 - y_i f(x_i))$ is the loss function; the parameter C determines the tradeoff between increasing the margin-size and ensuring the x_i does not violate the margin constraint.

REFERENCES

Abbott, M.B., J.C. Bathurst, J.A. Cunge, and P.E. O'connell, 1986. An Introduction to the European Hydrological System - Systeme Hydrologique Europeen (SHE), 1: History and Philosophy of a Physically-Based, Distributed Modelling. Journal of Hydrology. DOI:10.1016/0022-1694(86)90115-0.

Abrahart, R.J. and L.M. See, 2007. Neural Network Modelling of Non-Linear Hydrological Relationships. Hydrology and Earth System Sciences 11:1563-1579.

Abrahart, R.J., F. Anctil, P. Coulibaly, C.W. Dawson, N.J. Mount, L.M. See, A.Y. Shamseldin, D.P. Solomatine, E. Toth, and R.L. Wilby, 2012. Two Decades of Anarchy? Emerging Themes and Outstanding Challenges for Neural Network River Forecasting. Progress in Physical Geography 36:480-513.

Almoradie, A., A. Jonoski, and I. Popescu, 2013. Web Based Access to Water Related Data Using OGC WaterML 2.0. International Journal of Advanced Computer Science and Applications.

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 and Software 37:146-156.

Andrea Zimmer, B.M.A.S.A.A.O., 2010. Evolutionary Algorithm Memory Enhancement for Real-Time CSO Control. World Environmental and Water Resources Congress 2010:2251-2259.

Bae, D.H., D.M. Jeong, and G. Kim, 2007. Monthly Dam Inflow Forecasts Using Weather Forecasting Information and Neuro-Fuzzy Technique. Hydrological Sciences Journal. doi:10.1623/hysj.52.1.99.

Bajcsy, P., R. Kooper, and L. Marini, 2005. A Meta-Workflow Cyber-Infrastructure System Designed for Environmental Observatories. Technical Report: NCSA Cyber-Environments Divisions, ISDAO1-2005.

Ben-Hur, A. and J. Weston, 2010. A User's Guide to Support Vector Machines. Data Mining Techniques for the Life Sciences 609:223-239.

Bowden, G.J. and H.R. Maier, 2012. Real-Time Deployment of Artificial Neural Network Forecasting Models: Understanding the Range of Applicability. Water Resources Research. doi:10.1029/2012WR011984.

Breiman, L. and P. Spector, 1992b. Submodel Selection and Evaluation in Regression. the X-Random Case. International Statistical Review 60(3): 291-319.

Breiman, L., J.H. Friedman, R.A. Olshen, and C.J. Stone, 1984. Classification and Regression Trees. Wadsworth & Brooks. Monterey.

Cai, X., R. Zeng, W.H. Kang, J. Song, and A.J. Valocchi, 2015. Strategic Planning for Drought Mitigation Under Climate Change. Journal of Water Resources Planning and Management 141(9): 04015004-1-10.

Caruana, R. and A. Niculescu-Mizil, 2006. An Empirical Comparison of Supervised Learning Algorithms. ACM, New York, New York, USA.

Chang, L.-C., F.-J. Chang, K.-W. Wang, and S.-Y. Dai, 2010. Constrained Genetic Algorithms for Optimizing Multi-Use Reservoir Operation. Journal of Hydrology 390:66-74.

Cornish, C.R., C.S. Bretherton, and D.B. Percival, 2006. Maximal Overlap Wavelet Statistical Analysis with Application to Atmospheric Turbulence. Boundary-Layer Meteorology 119:339-374.

Cortes, C. and V. Vapnik, 1995. Support-Vector Networks. Machine Learning 20:273–297.

Coulibaly, P., F. Anctil, and B. Bobee, 2000. Daily Reservoir Inflow Forecasting Using Artificial Neural Networks with Stopped Training Approach. Journal of Hydrology 230:244–257.

Cunge, J.A., 1969. On the Subject of a Flood Propagation Computation Method (Musklngum Method). Journal of Hydraulic Research 7:205–230.

David, C.H., D.R. Maidment, G.-Y. Niu, Z.-L. Yang, F. Habets, and V. Eijkhout, 2011. River Network Routing on the NHDPlus Dataset. Journal of Hydrometeorology 12:913-934.

DE VOS, N.J. and T.H.M. RIENTJES, 2007. Multi-Objective Performance Comparison of an Artificial Neural Network and a Conceptual Rainfall-Runoff Model. Hydrological Sciences Journal 52:397-413.

Deb, K., 2000. An Efficient Constraint Handling Method for Genetic Algorithms. Computer Methods in Applied Mechanics and Engineering. doi:10.1016/S0045-7825(99)00389-8.

Dietterich, T.G., 2000. Ensemble Methods in Machine Learning. Multiple Classifier Systems, Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, Berlin, Heidelberg, pp. 1-15.

Dong, C., G. Schoups, and N. van de Giesen, 2013. Scenario Development for Water Resource Planning and Management: A Review. Technological Forecasting & Social Change 80:749-761.

Ebrahim, G.Y., A. Jonoski, and A. Al-Maktoumi, 2015. Simulation-Optimization Approach for Evaluating the Feasibility of Managed Aquifer Recharge in the Samail Lower Catchment, Oman. Journal of Water Planning and Management. doi:10.1061/(ASCE)WR.1943-5452.0000588.

El-Shafie, A., M.R. Taha, and A. Noureldin, 2006. A Neuro-Fuzzy Model for Inflow Forecasting of the Nile River at Aswan High Dam. Water Resources Management 21:533-556.

Elith, J., J.R. Leathwick, and T. Hastie, 2008. A Working Guide to Boosted Regression Trees. Journal of Animal Ecology 77:802-813.

Erdal, H.I. and O. Karakurt, 2013. Advancing Monthly Streamflow Prediction Accuracy of CART Models Using Ensemble Learning Paradigms. Journal of Hydrology 477:119–128.

Fang, S., L. Xu, H. Pei, Y. Liu, Z. Liu, Y. Zhu, J. Yan, and H. Zhang, 2013. An Integrated Approach to Snowmelt Flood Forecasting in Water Resource Management. IEEE Transactions on Industrial Informatics 10:548-558.

Friedman, J.H., 2001. Greedy Function Approximation: A Gradient Boosting Machine. Annals of Statistics. doi:10.2307/2699986.

Gaur, S., B.R. Chahar, and D. Graillot, 2011. Analytic Elements Method and Particle Swarm Optimization Based Simulation-Optimization Model for Groundwater Management. Journal of Hydrology.

Geller, G.N. and W. Turner, 2007. The Model Web: a Concept for Ecological Forecasting. Geoscience and Remote Sensing Symposium. 2469-2472.

Georgakopoulos, D., M. Hornick, and A. Sheth, 1995. An Overview of Workflow Management: From Process Modeling to Workflow Automation Infrastructure. Distributed and Parallel Databases 3:119-153.

Goldberg, D.E. and J.H. Holland, 1988. Genetic Algorithms and Machine Learning. Machine Learning.

Goodall, J.L., J.S. Horsburgh, and T.L. Whiteaker, 2008. A first approach to web services for the National Water Information System. Environmental Modelling & Software 23: 404-411

Gu, J., G.Y. Li, and Z. Dong, 2012. Hybrid and Adaptive Meta-Model-Based Global Optimization. Engineering Optimization 44:87-104.

Hamarat, C., J.H. Kwakkel, E. Pruyt, and E.T. Loonen, 2014. An Exploratory Approach for Adaptive Policymaking by Using Multi-Objective Robust Optimization. Simulation Modelling Practice and Theory 46:25-39.

Hastie, T., R. Tibshirani, and J. Friedman, 2008. Boosting and Additive Trees. The Elements of Statistical Learning, Springer Series in Statistics. Springer New York, New York, NY, pp. 1-51.

Herrera, F., M. Lozano, and J.L. Verdegay, 1998. Tackling Real-Coded Genetic Algorithms: Operators and Tools for Behavioural Analysis. Artificial Intelligence Review 12:265-319.

Hsu, C.W., C.C. Chang, and C.J. Lin, 2003. A Practical Guide to Support Vector Classification.

Huffaker, R., N. Whittlesey, and J.R. Hamilton, 2000. The Role of Prior Appropriation in Allocating Water Resources Into the 21st Century. International Journal of Water Resources Development 16:265-273.

Hydrologic Engineering Center, 2011. Accelerated Corps Water Management System (CWMS) Deployment Campaign. :1-98.

Islam, Z., 2015. A Review on Physically Based Hydrologic Modeling. https://www.researchgate.net/publication/272169378_A_Review_on_Physically_Based_Hydrologic_Modeling, *accessed* December 2015

Jain, S.K., A. Das, and D.K. Srivastava, 1999. Application of ANN for Reservoir Inflow Prediction and Operation. Journal of Water Resources. DOI:10.1061/(ASCE)0733-9496(1999)125:5(263).

John, H.H., 1975. Adaptation in Natural and Artificial Systems: an Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence. USA: University of Michigan.

Johnson, V.M. and L.L. Rogers, 2000. Accuracy of Neural Network Approximators in Simulation-Optimization. Journal of Water Resources Planning and Management

Jones, D., N. Jones, J. Greer, and J. Nelson, 2015. A Cloud-Based MODFLOW Service for Aquifer Management Decision Support. Computers and Geosciences 78:81-87.

Jothiprakash, V. and R.B. Magar, 2012. Multi-Time-Step Ahead Daily and Hourly Intermittent Reservoir Inflow Prediction by Artificial Intelligent Techniques Using Lumped and Distributed Data. Journal of Hydrology 450-451:293-307.

Kaini, P., K. Artita, and J.W. Nicklow, 2012. Optimizing Structural Best Management Practices Using SWAT and Genetic Algorithm to Improve Water Quality Goals. Water Resources Management 26:1827-1845.

Kang, B., Y.H. Ku, and Y. Do Kim, 2015. A Case Study for ANN-Based Rainfall–Runoff Model Considering Antecedent Soil Moisture Conditions in Imha Dam Watershed, Korea. Environmental Earth Sciences:1-12.

Kang, D. and K. Lansey, 2013. Scenario-Based Robust Optimization of Regional Water and Wastewater Infrastructure. Journal of Water Resources Planning and Management 139:325–338.

Kohavi, R., 1995. A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection.

Kratica, J., 1999. Improving Performances of the Genetic Algorithm by Caching. Computers and Artificial Intelligence.

Kumar, S., M.K. Tiwari, C. Chatterjee, and A. Mishra, 2015. Reservoir Inflow Forecasting Using Ensemble Models Based on Neural Networks, Wavelet Analysis and Bootstrap Method. Water Resources Management 29:4863-4883.

Laniak, G.F., G. Olchin, J. Goodall, A. Voinov, M. Hill, P. Glynn, G. Whelan, G. Geller, N. Quinn, M. Blind, S. Peckham, S. Reaney, N. Gaber, R. Kennedy, and A. Hughes, 2013. Integrated Environmental Modeling: a Vision and Roadmap for the Future. Environmental Modelling and Software 39:3-23.

Lee, S., T.D. Wang, N. Hashmi, and M.P. Cummings, 2007. Bio-STEER: a Semantic Web Workflow Tool for Grid Computing in the Life Sciences. Future Generation Computer Systems 23:497-509.

Maier, H.R. and G.C. Dandy, 2000. Neural Networks for the Prediction and Forecasting of Water Resources Variables: a Review of Modelling Issues and Applications. Environmental Modelling and Software 15:101-124.

Maier, H.R., A. Jain, G.C. Dandy, and K.P. Sudheer, 2010. Methods Used for the Development of Neural Networks for the Prediction of Water Resource Variables in River Systems: Current Status and Future Directions. Environmental Modelling and Software 25:891–909.

Maier, H.R., Z. Kapelan, J. Kasprzyk, J. Kollat, L.S. Matott, M.C. Cunha, G.C. Dandy, M.S. Gibbs, E. Keedwell, A. Marchi, A. Ostfeld, D. Savic, D.P. Solomatine, J.A. Vrugt, A.C.

Zecchin, B.S. Minsker, E.J. Barbour, G. Kuczera, F. Pasha, A. Castelletti, M. Giuliani, and P.M. Reed, 2014. Evolutionary Algorithms and Other Metaheuristics in Water Resources: Current Status, Research Challenges and Future Directions. Environmental Modelling and Software 62:271-299.

Marini, L., R. Kooper, and P. Bajcsy, 2007. Supporting Exploration and Collaboration in Scientific Workflow Systems. AGU Fall Meeting.

Martin, D.M., D. Harrison Atlas, N.A. Sutfin, and N.L. Poff, 2014. A Social-Ecological Framework to Integrate Multiple Objectives for Environmental Flows Management. Journal of Contemporary Water Research & Education 153:49-58.

Mateo, C.M., N. Hanasaki, D. Komori, K. Tanaka, M. Kiguchi, A. Champathong, T. Sukhapunnaphan, D. Yamazaki, and T. Oki, 2014. Assessing the Impacts of Reservoir Operation to Floodplain Inundation by Combining Hydrological, Reservoir Management, and Hydrodynamic Models. Water Resources Research 50:7245-7266.

McCarthy, G.T., 1938. The Unit Hydrograph and Flood Routing, Conf. North Atlantic Div.

McHenry, K., R. Kooper, M. Ondrejcek, L. Marini, and P. Bajcsy, 2011. A Mosaic of Software. IEEE, pp. 279-286.

Microsoft, A.M.T., 2015. AzureML: Anatomy of a Machine Learning Service. :1–9.

Mitchell, T.M., 1997. Machine Learning. 1997. Burr Ridge.

Nazari, S., S.J. Mousavi, K. Behzadian, and Z. Kapelan, 2014. Sustainable Urban Water Management: a Simulation Optimization Approach. The International Conference on Hydroinformatics.

Nicklow, J., P. Reed, D. Savic, and T. Dessalegne, 2009. State of the Art for Genetic Algorithms and Beyond in Water Resources Planning and Management. Journal of Water Planning and Management.

Nourani, V., M. Komasi, and A. Mano, 2009. A Multivariate ANN-Wavelet Approach for Rainfall–Runoff Modeling. Water Resources Management. doi:10.1007/s11269-009-9414-5.

Office, T.C., 2012. The Impact of the 2011 Drought and Beyond.

Oinn, T., M. Addis, J. Ferris, D. Marvin, M. Senger, M. Greenwood, T. Carver, K. Glover, M.R. Pocock, A. Wipat, and P. Li, 2004. Taverna: a Tool for the Composition and Enactment of Bioinformatics Workflows. Bioinformatics 20:3045-3054.

Okkan, U., 2012. Wavelet Neural Network Model for Reservoir Inflow Prediction. Scientia Iranica 19:1445-1455.

PALLOTTINO, S., G. SECHI, and P. ZUDDAS, 2005. A DSS for Water Resources Management Under Uncertainty by Scenario Analysis. Environmental Modelling and Software 20:1031-1042.

Pasha, M. and K. Lansey, 2010. Strategies for Real Time Pump Operation for Water Distribution Systems. Water Distribution Systems Analysis 2010. doi:10.1061/41203(425)130.

Percival, D.B. and A.T. Walden, 1993. Spectral Analysis for Physical Applications. Cambridge University Press.

Polikar, R., 2001. The Wavelet Tutorial Part I. Fundamental Concepts and an Overview of the Wavelet Theory.

Rasekh, A. and K. Brumbelow, 2015. A Dynamic Simulation-Optimization Model for Adaptive Management of Urban Water Distribution System Contamination Threats. Applied Soft Computing Journal 32:59-71.

Razavi, S., B.A. Tolson, and D.H. Burn, 2012a. Numerical Assessment of Metamodelling Strategies in Computationally Intensive Optimization. Environmental Modelling and Software 34:67-86.

Razavi, S., B.A. Tolson, and D.H. Burn, 2012b. Review of Surrogate Modeling in Water Resources. Water Resources Research.

Richter, B.D., R. Mathews, D.L. Harrison, and R. Wigington, 2003. Ecologically Sustainable Water Management: Managing River Flows for Ecological Integrity. Ecological Applications 13:206–224.

Roman, D., S. Schade, and A.J. Berre, 2009. Model as a Service (MaaS). AGILE Workshop: Grid Technologies for Geospatial Applications.

Rui, H. and D. Mocko, 2013. Readme Document for North America Land Data Assimilation System Phase 2 (NLDAS-2) Products. Greenbelt.

Sharma, A. and R. Mehrotra, 2014. An Information Theoretic Alternative to Model a Natural System Using Observational Information Alone. Water Resources Research. doi:10.1002/2013WR013845.

Sharma, A., R. Mehrotra, J. Li, and S. Jha, 2016. A Programming Tool for Nonparametric System Prediction Using Partial Informational Correlation and Partial Weights. Environmental Modelling and Software 83:271-275.

Singh, A., 2013. Irrigation Planning and Management Through Optimization Modelling. Water Resources Management 28:1-14.

Singh, V.P. and D.A. Woolhiser, 2002. Mathematical Modeling of Watershed Hydrology. Journal of Hydrologic Engineering.

Snelder, T.H., N. Lamouroux, J.R. Leathwick, and H. Pella, 2009. Predictive Mapping of the Natural Flow Regimes of France. Journal of Hydrology, 373(1), 57-67.

Solomatine, D.P. and A. Ostfeld, 2008. Data-Driven Modelling: Some Past Experiences and New Approaches. Journal of Hydroinformatics 10:3-20.

Stepney, L., 2012. Texas Commision on Environmental Quality.

Sun, A., 2013. Enabling Collaborative Decision-Making in Watershed Management Using Cloud-Computing Services. Environmental Modelling and Software 41:93–97.

Sun, A.Y., R.M. Miranda, and X. Xu, 2014. Development of Multi-Metamodels to Support Surface Water Quality Management and Decision Making. Environmental Earth Sciences 73:423-434.

Tabari, M.M.R. and J. Soltani, 2012. Multi-Objective Optimal Model for Conjunctive Use Management Using SGAs and NSGA-II Models. Water Resources Management 27:37–53.

Tao, T., 1999. Local Inflow Calculator for Reservoirs. Canadian Water Resources Journal. doi:10.4296/cwrj2401053.

Thompson, J.R.B.A.F.E., 2012. Simulation of Streamflow and the Effects of Brush Management on Water Yields in the Upper Guadalupe River Watershed, South-Central Texas, 1995-2010 :1-36.

Toukourou, M., A. Johannet, G. Dreyfus, and P.-A. Ayral, 2010. Rainfall-Runoff Modeling of Flash Floods in the Absence of Rainfall Forecasts: the Case of "Cévenol Flash Floods." Applied Intelligence 35:178-189.

Valens, C., 1999. A Really Friendly Guide to Wavelets. DOWNLOAD DA INTERNET: https://www.cs.unm.edu/~williams/cs530/arfgtw.pdf, *accessed* December 2015

Valipour, M., M.E. Banihabib, and S.M.R. Behbahani, 2013. Comparison of the ARMA, ARIMA, and the Autoregressive Artificial Neural Network Models in Forecasting the Monthly Inflow of Dez Dam Reservoir. Journal of Hydrology 476:433-441.

Viana, F.A.C., R.T. Haftka, and L.T. Watson, 2012. Efficient Global Optimization Algorithm Assisted by Multiple Surrogate Techniques. Journal of Global Optimization 56:669-689.

Villa, F., I.N. Athanasiadis, and A.E. Rizzoli, 2009. Modelling with Knowledge: a Review of Emerging Semantic Approaches to Environmental Modelling. Environmental Modelling and Software 24:577-587.

Watkins, D.W., Jr and D.C. McKinney, 1997. Finding Robust Solutions to Water Resources Problems. Journal of Water Resources Planning and Management 123(1):49-58

Wilhite, D.A. and M.H. Glantz, 2009. Understanding: the Drought Phenomenon: the Role of Definitions. Water International 10:111-120.

Wright, A.H., 1991. Genetic Algorithms for Real Parameter Optimization. Foundations of Genetic Algorithms. Elsevier, pp. 205-218.

Wu, B., Y. Zheng, X. Wu, Y. Tian, F. Han, J. Liu, and C. Zheng, 2015. Optimizing Water Resources Management in Large River Basins with Integrated Surface Water-Groundwater Modeling: a Surrogate-Based Approach. Water Resources Research 51:2153-2173.

Yan, S. and B. Minsker, 2011. Applying Dynamic Surrogate Models in Noisy Genetic Algorithms to Optimize Groundwater Remediation Designs. Journal of Water Resources Planning and Management 137:284-292.

Yeniay, O., 2005. Penalty Function Methods for Constrained Optimization with Genetic Algorithms. Mathematical and Computational Applications.