Article type : Technical Paper

# Statistical and Hybrid Methods Implemented in a Web Application for Predicting Reservoir Inflows During Flood Events

Tingting Zhao, Barbara Minsker, Fernando Salas, David Maidment, Vesselin Diev, Jacob Spoelstra, and Prashant Dhingra

Graduate Student (**Zhao**), Department of Civil and Environmental Engineering, University of Illinois at Urbana-Champaign, 4129 Newmark Civil Engineering Laboratory, Urbana, IL 61801; Professor (**Minsker**), Department of Civil and Environmental Engineering, Southern Methodist University, Dallas, TX 75275; Associate Scientist (**Salas**), UCAR, NOAA/NWS National Water Center, Tuscaloosa, AL 35401; Professor (**Maidment**), Center for Research in Water Resources, University of Texas at Austin, Austin, TX 78712; and Senior Data Scientist (**Diev**), Microsoft Azure Data Science Team, Director of Data Science (**Spoelstra**), Azure Machine Learning Platform, and Principle Project Manager (**Dhingra**), Data Science and Machine Learning, Microsoft, Redmond, WA 98052 (E-Mail/Zhao: tzhao6@illinois.edu).

**Abstract:** 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. This study's objective is to build real-time data-driven services that enable managers to rapidly estimate reservoir inflows from available data and models. 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 study demonstrates 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. 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.

(**Key Terms**: flooding; data-driven model services; AzureML; reservoir inflow.)

## INTRODUCTION

In this paper we demonstrate a new framework for real-time flood management through data-driven services to rapidly estimate reservoir inflows from available data and models. 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-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). This paper explores these alternative approaches for real-time flood management and implements the resulting models as real-time services using the AzureML service. More background for each of these components is given below.

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 (Singh & Woolhiser, 2011).

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 analyze relationships between concurrent input and output time series (Solomatine and Ostfeld, 2008) and can be applied either alone (using a purely statistical model) or in conjunction with physics-based models, which use mathematical equations derived from the physical processes (creating a hybrid model). Machine learning methods are a type of statistical approach that can be fit rapidly and automatically to represent highly complex relationships. The popular data-driven machine learning methods used in river systems include artificial neural networks (ANN), fuzzy rule-based systems, and support vector machines (SVM), among others.

Hybrid models often use machine learning approaches to fit error models to the residual errors in a physics-based forecasting model. Gragne et al. (2015) implemented a filter updating procedures to update error forecast to improve reservoir inflow forecasts. Gragne et al. (2015) proposed an error model to improve hourly reservoir inflow forecasts over one day ahead.

Many applications of ANN focus on rainfall-runoff models (e.g., Sharma et al., 2000, 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 inflows. Bae et al. (2007) implemented Adaptive Neuro-Fuzzy Inference System (ANFIS) to predict monthly dam inflow using past observed data and future weather forecasting information. Zhang et al. (2009) implemented multilayer perceptron artificial neural networks (MLP-ANNs) using observed precipitation and forecasted precipitation from QPF to predict daily reservoir inflow. Sharma and Chowdhury (2011) reviewed static and dynamic ensemble methods in probabilistic reservoir system forecasting models to reduce structural errors. Jothiprakash and Magar (2012) predicted daily and hourly intermittent rainfall and reservoir inflow using ANN, an adaptive neuro-fuzzy inference system (ANFIS), and linear genetic programming (GLP). Valipour (2013) compared autoregressive moving average (ARMA) and autoregressive integrated moving average (ARIMA) using increasing number 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 forecast instead of point predictions for reservoir inflow.

These previous studies have focused on non-linear regression models and the predictive performance is good when compared 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.

As described above, Artificial Neural Network (ANN) is a commonly used data-driven approach in hydrology (see also Bowden et al., 2012, Abrahart et al., 2012, Maier et al., 2010), but the convergence speed is low and training can require significant time that may be a barrier when near-real-time model updating is required (e.g., Jain et al., 1999, Maier & Dandy, 2000). This study uses boosted regression trees (BRT) as function approximators. Our early tests showed that BRT has advantages in faster training and higher accuracy than ANN for this application. 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, which 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 (creating ensembles of multiple models that combine fast but weak learners to create a strong leaner). The approach combines multiple simple trees into an additive regression model to improve predictive performance (Elith et al. 2008).

The data-driven models in this study were developed using AzureML Studio Predictive Analytics, a Cloud-hosted user-friendly software toolkit that allows graphical construction of data analysis steps ("workflows") such as data requests, fitting data-driven models, and data visualization (AzureML team Microsoft, *2015*). The data-driven models built in AzureML Studio can be published as Web services on the Azure Cloud, providing scalability and high software availability and reliability, as well as easy integration into modern software systems.

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 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.

## METHODOLOGY

Figure 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 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, $\boldsymbol{y}=f(\boldsymbol{x})$ is a mapping (training model) between input variables $\mathbf{x}$ and output variable $\mathbf{y}$. 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 modelthat 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.

**[Insert Fig 1 here]**

*Data Preprocessing Using Wavelet Analysis*

Wavelet analysis is used to filter the reservoir inflow data into trend and noise parts. This step is necessary because there are no direct measures of reservoir inflows. Reservoir inflows are derived from reservoir storage and *flowout* (the flow out from the reservoir), which are subject to fluctuations that may be caused by wave action when winds are high during storms or by measurement errors of the sensors at the gauging stations (Tao, 1998). 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 since 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 wavelet coefficient generated by a high-pass filter is defined as

$$W_{j,t} = \sum_{l=0}^{L_j-1} \tilde{h}_{j,l} X_{t-l} \quad (1)$$

The wavelet coefficient generated by a low-pass filter is defined as

$$V_{j,t} = \sum_{l=0}^{L_j-1} \tilde{g}_{j,l} X_{t-l} \quad (2)$$

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

The decomposition process is shown in Figure 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 $V_1$ and $W_1$. The level 2 decomposition is based on $V_1$, which is the trend component from the last level. $W_1$ 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.

**[Insert Fig 2 here]**

*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 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 paper. Key features are the ability to fit complex nonlinear models and high accuracy (Elith et al., 2008, Caruana & Niculescu-Mizil, 2006).

**[Insert Fig 3 here]**

*Performance Metrics*

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

a. Mean Absolute Error (MAE)

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|\hat{y}_i - y_i| \quad (3)$$

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)

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2} \quad (4)$$

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)

$$RAE = \frac{\sum_{i=1}^{n}|\hat{y}_i - y_i|}{\sum_{i=1}^{n}|y_i - \bar{y}|} \quad (5)$$

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)

$$RSE = \frac{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \quad (6)$$

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 ($R^2$)

$$R^2 = \left(\frac{\sum_{i=1}^{n}(\hat{y}_i - \bar{\hat{y}})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(\hat{y}_i - \bar{\hat{y}})^2 \sum_{i=1}^{n}(y_i - \bar{y})^2}}\right)^2 \quad (7)$$

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.

*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 4. Such data workflows, including data preprocessing, model building, and results visualization, are more natural and intuitive than scripts. Non-technical users can implement and update the data-driven approach without requiring advanced 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 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.

**[Insert Fig 4 here]**

## CASE STUDY

Lake Travis is in Travis County, located upstream of Lake Austin. Mansfield Dam, operated by LCRA, creates Lake Travis, which serves to contain floodwaters and helps to manage flooding downstream. 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).

**[Insert Fig 5 here]**

*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 6. Precipitation and soil moisture input data were collected from 31 grid points in Lake Travis Basin in the upstream of Mansfield Dam, as shown in Figure 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 5.a.

**[Insert Fig 6 here]**

The precipitation data (in kg/m$^2$) 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 1/8$^{th}$ degree grid spacing (Rui & Mocko, 2013). The soil moisture data, in units of kg/m$^2$, relied on the Noah land surface model (Noah soil moisture 0-100 cm). Data from both models can be 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.

*Model Implementation*

**Wavelet analysis to filter data noise**.

Wavelet analysis is intended to smooth the fluctuations in the reservoir inflow data and keep the trend. The decomposition level (Figure 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:

$$L = int[log_{10}\,(N)] \tag{8}$$

where $L$ is the decomposition level and $N$ is the number of data values.

In this study, the number of time series data is 1656. Based on Equation 8, the decomposition level $L$ = int[log(1656)] = 3. To select the best decomposition level, Figure 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 7.b, with level 2 noise removal, represents the best data filtering: the dataset is smooth and the peak is not excessively truncated. Figure 8 shows the original reservoir inflow versus the filtered reservoir inflow.

**[Insert Fig 7 here]**

**[Insert Fig 8 here]**

**Correlation.**

To assess appropriate time lags for inclusion in the model, cross correlation was performed and the results are shown in Figure 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.

**[Insert Fig 9 here]**

Figure 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 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 to identify which combinations of variables are most predictive. The variables used in the best performing models were those that decreased errors the most.  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 that is 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 that is 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 that is 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 that is 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 that is 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 that is 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 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 11 shows the structure of one example regression tree in the BRT. The algorithm takes the entire data set as an input and then splits the dataset at the value of one feature variable ("node") that maximizes the "separation" of the dataset. Separation is measured by the variance reduction, shown in Equation (9), which measures the total reduction in variance of the output variable due to the split of the node (Timofeev, 2004, Breiman, 1984). Selecting the feature variable with the largest variance reduction minimizes the model error at each split (Timofeev, 2004). The dataset is then split into two parts based on this value (in Figure 11, the value for the first split is flowin_lag ≤ 36965 at the root node). Similar splitting continues, as shown in Figure 11, until the stopping criterion (the maximum number of leaves per tree=10 in this case) is met. Finally, a prediction value of the output is obtained at each leaf node of the tree. For instance, if the flowin_lag >36965 and smLoc24>357.4, the prediction value is 4469.

$$I_V(N) = \frac{1}{|S|^2}\sum_{i\in S}\frac{1}{2}(y_i-\widehat{c_0})^2 - (\frac{1}{|S_t|^2}\sum_{i\in S_t}\frac{1}{2}(y_i-\widehat{c_1})^2 + \frac{1}{|S_f|^2}\sum_{i\in S_f}\frac{1}{2}(y_i-\widehat{c_2})^2 \qquad (9)$$

where $S$ is the original dataset, $S_t$ and $S_f$ are the split datasets, and $\widehat{c_0}$, $\widehat{c_1}$, $\widehat{c_2}$ represent the estimate of the average of output label in the respective dataset.

**[Insert Fig 10 here]**

**[Insert Fig 11 here]**

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

Figure 12 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.

**[Insert Fig 12 here]**

## RESULTS

*Physics-based Model Performance*

Figure 13 shows the predicted reservoir inflow from the physics-based model HEC-HMS in CWMS and Table 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.

**[Insert Fig 13 here]**

**[Insert Table 1 here]**

*Data-Driven Flow Prediction Model*

Table 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 14, 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 2 and 3. Figure 15 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.

**[Insert Table 2 here]**

**[Insert Fig 14 here]**

**[Insert Fig 15 here]**

*Hybrid 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 12. The predicted reservoir inflow is then calculated using the predicted residual error plus the predicted reservoir inflow from the physics-based model. Table 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 16 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 14).

Figure 17 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.

**[Insert Table 3 here]**

**[Insert Fig 16 here]**

**[Insert Fig 17 here]**

*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 18). Figure 18.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 18.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 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 18.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.

**[Insert Fig 18 here]**

## DISCUSSION AND CONCLUSIONS

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 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 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, 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 12, 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 19). 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 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.

**[Insert Fig 19 here]**

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.

## APPENDIX

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 $\epsilon$ 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 $F_t(x)$


## ACKNOWLEDGMENTS

The authors are grateful to the 2015 National Flooding Interoperability Experiment (NFIE) Summer Institute for providing an opportunity to conduct this study. The authors also acknowledge Microsoft Research for funding support, the Microsoft Azure data science team for technical support, and LCRA staff David Walker and Chris Riley for providing data and expertise on the case study.


## LITERATURE CITED


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(5):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.

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 52(1): 99-113. DOI:10.1623/hysj.52.1.99.

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 48(10). DOI:10.1029/2012WR011984.

Breiman, L., J. Friedman, C.J. Stone, and R.A. Olshen, 1984. Classification and Regression Trees.

Caruana, R. and A. Niculescu-Mizil, 2006. An Empirical Comparison of Supervised Learning Algorithms. In Proceedings of the 23rd international conference on Machine learning, ACM, New York, USA, pp. 161-168

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.

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.

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.

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.

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

Gragne, A.S., K. Alfredsen, A. Sharma, and R. Mehrotra, 2015. Recursively Updating the Error Forecasting Scheme of a Complementary Modelling Framework for Improved Reservoir Inflow Forecasts. Journal of Hydrology. doi:10.1016/j.jhydrol.2015.05.039.

Gragne, A.S., A. Sharma, R. Mehrotra, and K. Alfredsen, 2015. Improving Real-Time Inflow Forecasting Into Hydropower Reservoirs Through a Complementary Modelling Framework. Hydrology and Earth System Sciences 19:3695–3714.

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.

Hydrologic Engineering Center, 2011. Accelerated Corps Water Management System (CWMS) Deployment Campaign. http://www.hec.usace.army.mil/publications/ProjectReports/PR-79.pdf, *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).

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 450:293-307.

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.

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.

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.

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.

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

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

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.

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

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.

Roncak, P., J. Bezak, and Z. Bajtek, 2012. Comparison of a Physically-Based and Data-Driven Model in the Bela River Basin. International Multidisciplinary Scientific GeoConference: SGEM: Surveying Geology & mining Ecology Management 3 (2012): 595.

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

Sharma, A. and S. Chowdhury, 2011. Coping with Model Structural Uncertainty in Medium-Term Hydro-Climatic Forecasting. Hydrology Research. doi:10.2166/nh.2011.104.

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.

Sharma, A., K.C. Luk, I. Cordery, and U. Lall, 2000. Seasonal to Interannual Rainfall Probabilistic Forecasts for Improved Water Supply Management: Part 2 - Predictor Identification of Quarterly Rainfall Using Ocean-Atmosphere Information. Journal of Hydrology. doi:10.1016/S0022-1694(00)00348-6.

Singh, V.P. and D.A. Woolhiser, 2002. Mathematical Modeling of Watershed Hydrology. Journal of hydrologic engineering, 7(4), 270-292

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.

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

Timofeev, R., 2004. Classification and Regression Trees (CART) Theory and Applications.

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.

Zhang, J., C. Cheng, S. Liao, and X. Wu, 2009. Daily Reservoir Inflow Forecasting Combining QPF Into ANNs Model. Hydrology and Earth System Sciences: 121-150

## TABLES

**TABLE 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 |

**TABLE 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 | $SM(t)_{closest}$,$SM(t-1)_{closest}$,$SM(t-2)_{closest}$,$Precip(t)_{closest}$,$Precip(t-1)_{closest}$, $Precip(t-2)_{closest}$, | flowin(t) | 27.666 | 52.953 | 0.160 | 0.048 | 0.953 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | flowout(t-2), flowin_lag(t-1), flowin_lag(t-2) | | | | | | |

**TABLE 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 |

LIST OF FIGURES

FIGURE 1. Framework of the Data-Driven Services

FIGURE 2. Decomposition Based on Wavelet Analysis

FIGURE 3. Example of a Regression Tree

FIGURE 4. An Example of AzureML Graphical Workflow

FIGURE 5. Case Study Location and Data Points

FIGURE 6. Inputs to and Outputs from the Data-driven Models

FIGURE 7. *Reservoir Inflow* Graph after Each Decomposition Level

FIGURE 8. Original Observed *Flowin* vs Filtered Observed *Flowin* During 2014-2015 Flooding Events

FIGURE 9. Correlation Plot between Input Features and Output Label

FIGURE 10. AzureML Workflow for Data-Driven Flow Prediction Model

FIGURE 11. Structure of an Example Regression Tree in BRT

FIGURE 12. Residual errors between Filtered Observed *Flowin* and *Flowin* from Physics-based Models during 2014-2015 Flooding Events

FIGURE 13. **Filtered** Observed *Flowin* vs Physics-based *Flowin* during 2014-2015 Flooding Events

FIGURE 14. **Filtered** Observed *Flowin* vs Predicted *Flowin* from May 26 2015 to Jun 5 2015

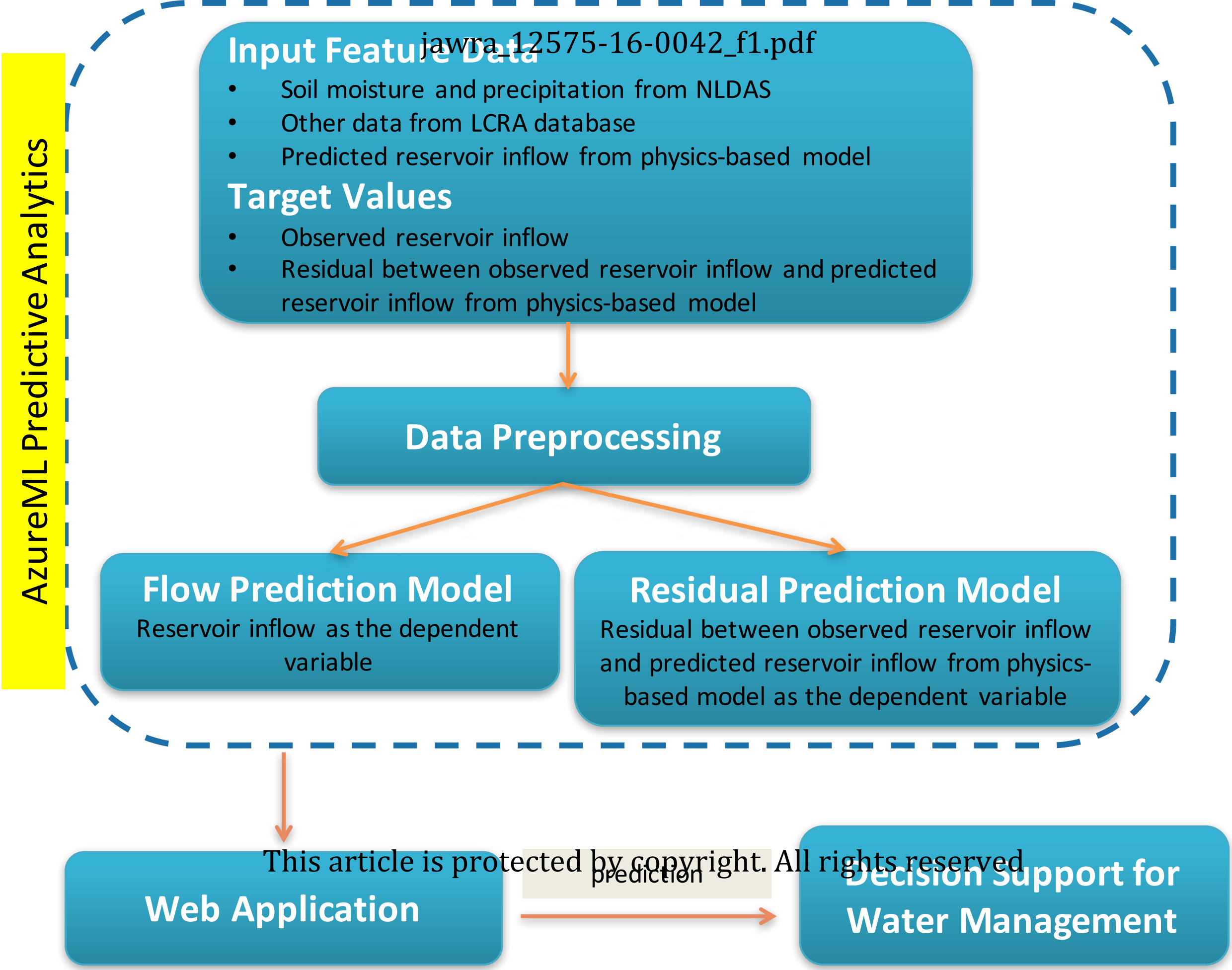
FIGURE 15. Flow Prediction Model Performance for Future Prediction

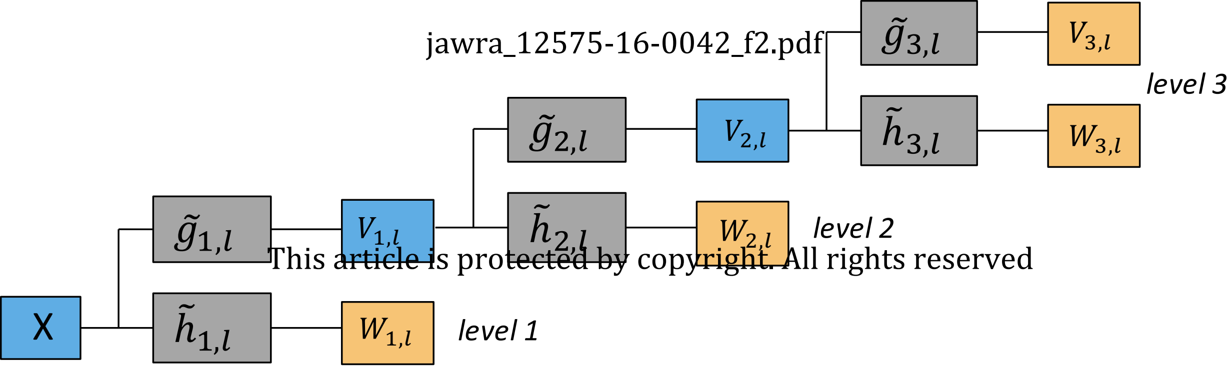
FIGURE 16. **Filtered** Observed *Flowin* vs Predicted *Flowin* from Hybrid Prediction Model vs Physics-based Model *Flowin* from May 26 2015 to June 5 2015

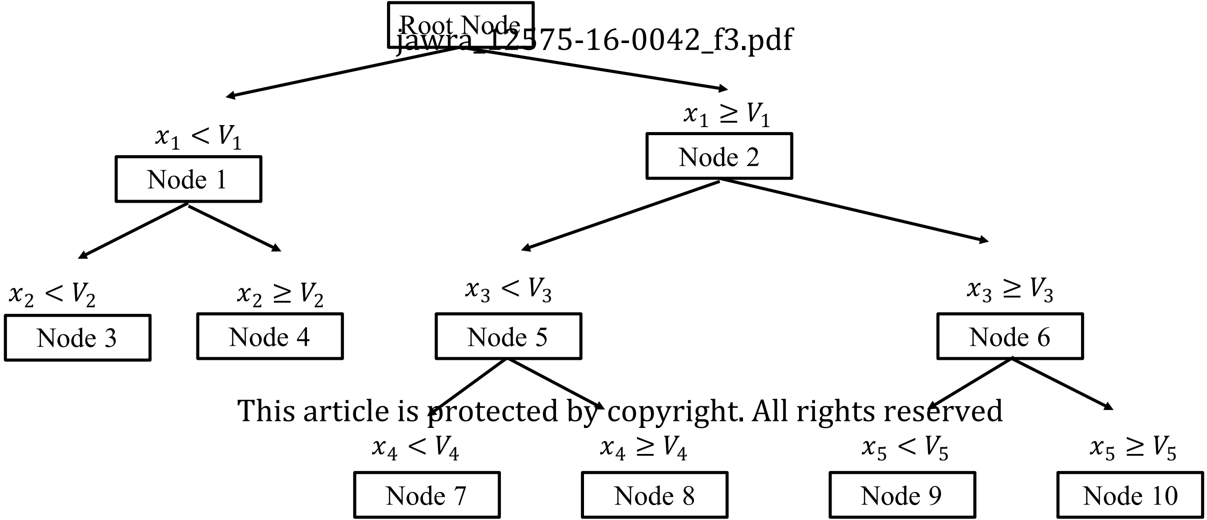
FIGURE 17. Hybrid Predicted Model Performance for Future Prediction

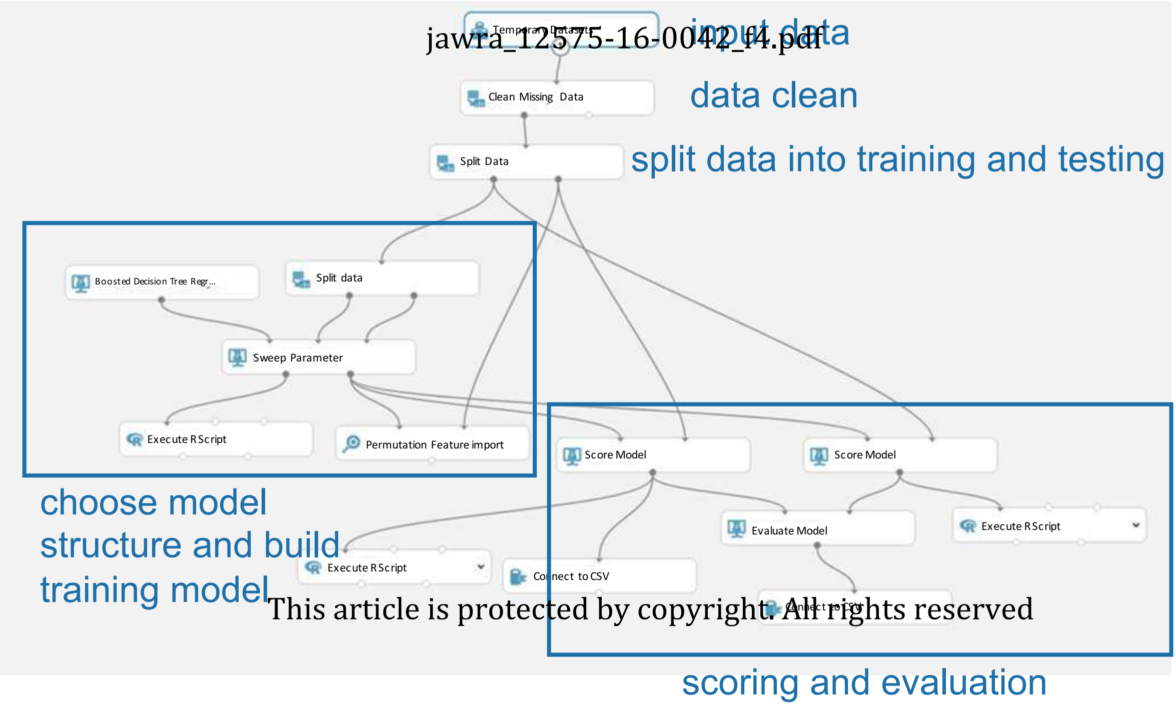
FIGURE 18. Screenshot of Web Interface for the Data-driven Model Services

FIGURE 19. Prediction Performance of Flow Prediction Model and Hybrid Prediction Model


AzureML Predictive Analytics
Input Feature Data
• Soil moisture and precipitation from NLDAS
• Other data from LCRA database
• Predicted reservoir inflow from physics-based model
Target Values
• Observed reservoir inflow
• Residual between observed reservoir inflow and predicted reservoir inflow from physics-based model
Data Preprocessing
Flow Prediction Model
Reservoir inflow as the dependent variable
Residual Prediction Model
Residual between observed reservoir inflow and predicted reservoir inflow from physics-based model as the dependent variable
Web Application
prediction
Decision Support for Water Management


$\tilde{g}_{3,l}$
$V_{3,l}$
level 3
$\tilde{g}_{2,l}$
$V_{2,l}$
$\tilde{h}_{3,l}$
$W_{3,l}$
$\tilde{g}_{1,l}$
$V_{1,l}$
$\tilde{h}_{2,l}$
$W_{2,l}$
level 2
X
$\tilde{h}_{1,l}$
$W_{1,l}$
level 1


Root Node
$x_1 < V_1$
Node 1
$x_1 \geq V_1$
Node 2
$x_2 < V_2$
Node 3
$x_2 \geq V_2$
Node 4
$x_3 < V_3$
Node 5
$x_3 \geq V_3$
Node 6
$x_4 < V_4$
Node 7
$x_4 \geq V_4$
Node 8
$x_5 < V_5$
Node 9
$x_5 \geq V_5$
Node 10


Temporary Datasets
input data
Clean Missing Data
data clean
Split Data
split data into training and testing
Boosted Decision Tree Regr...
Split data
Sweep Parameter
Execute R Script
Permutation Feature import
choose model structure and build training model
Score Model
Score Model
Evaluate Model
Execute R Script
Execute R Script
Connect to CSV
Connect to CSV
scoring and evaluation

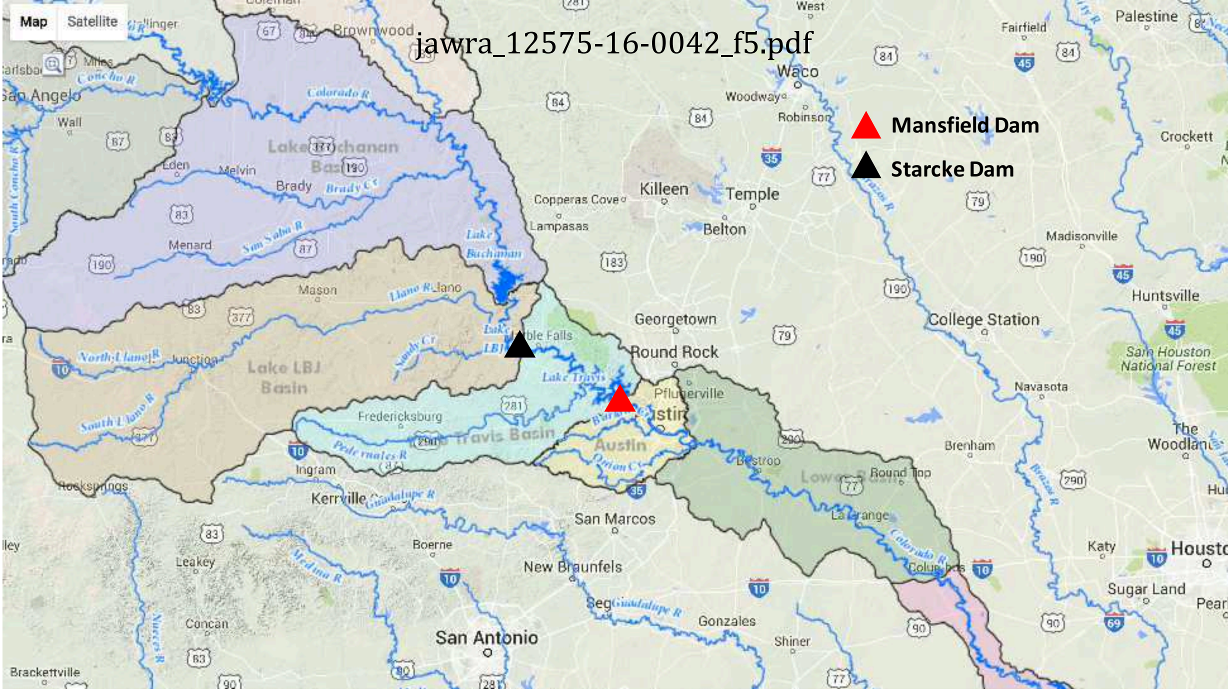



Figure 5.a Map of Lake Travis Basin

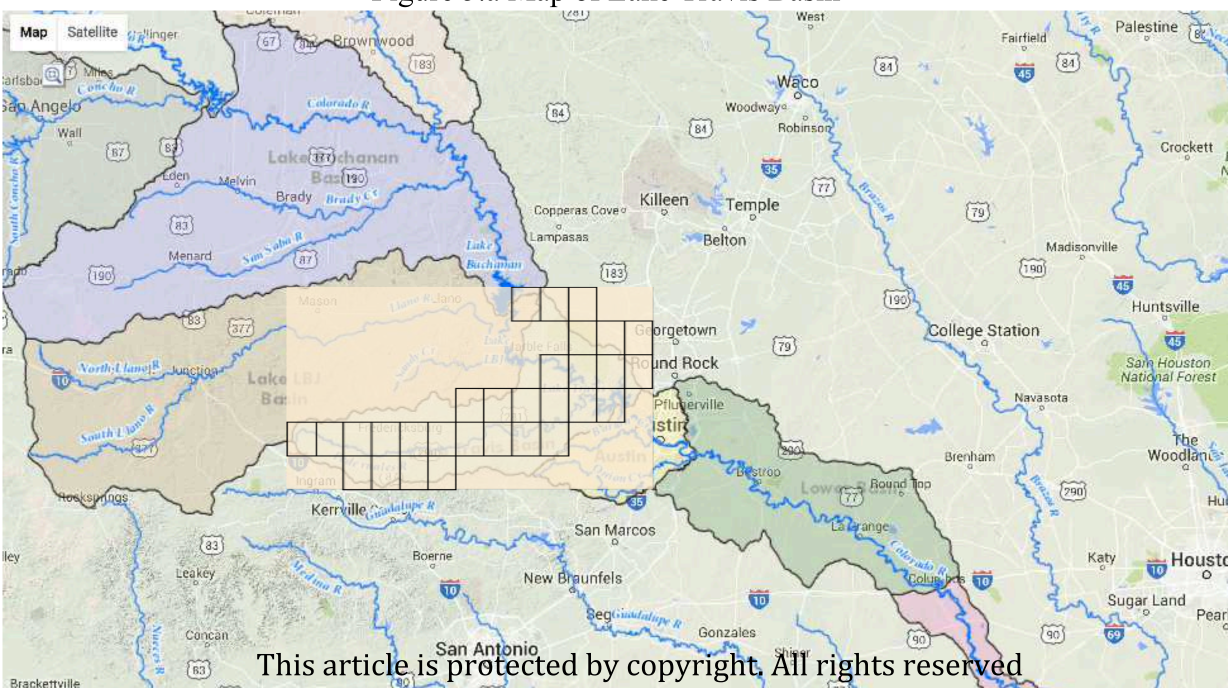




Figure 5.b Grid Points in Lake Travis Basin


Inputs
Soil moisture from NLDAS
Precipitation from NLDAS
Flow out of the upstream reservoir
Previous flowin to Mansfield Dam
Outputs
Reservoir inflow
Residual between observed reservoir inflow and predicted reservoir inflow from physics-based model

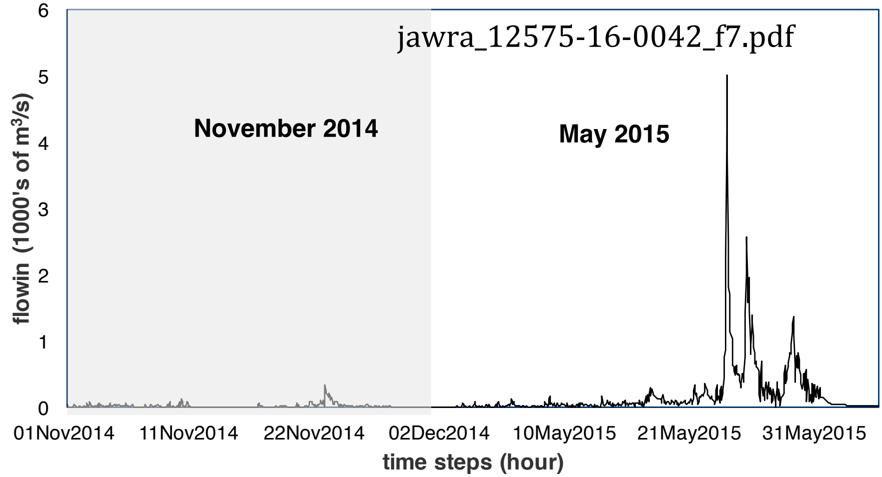



Figure 7.a. *Reservoir Inflow* at Level 1

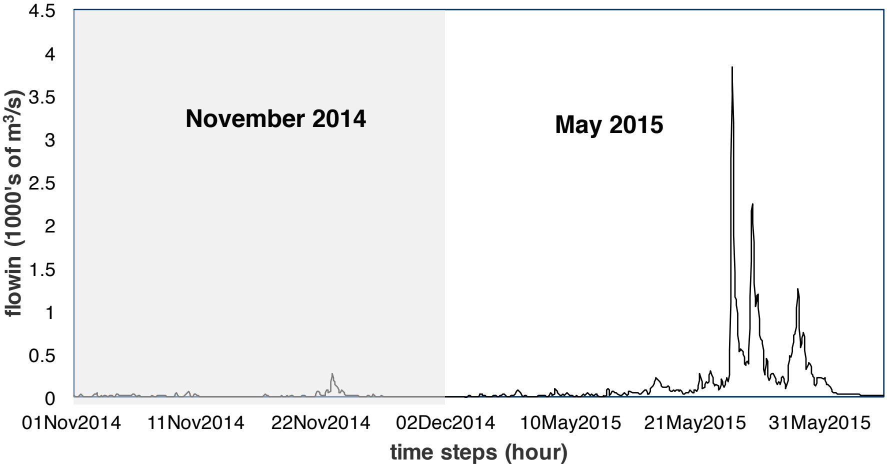



Figure 7.b. *Reservoir Inflow* at Level 2

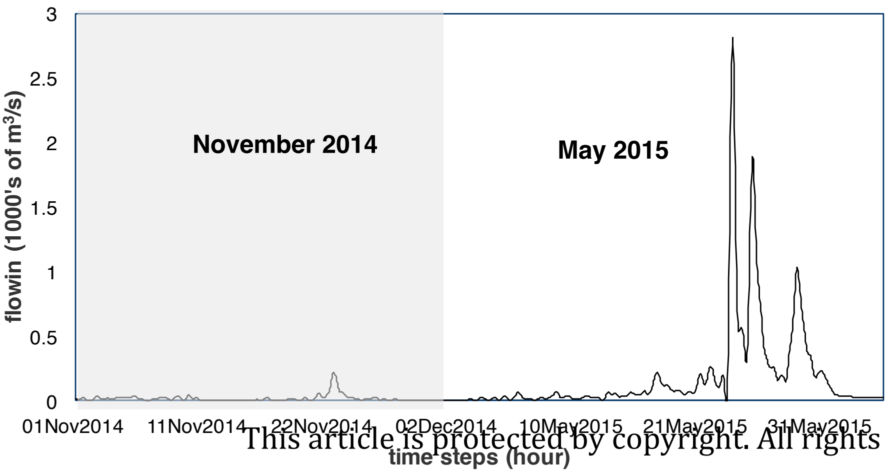
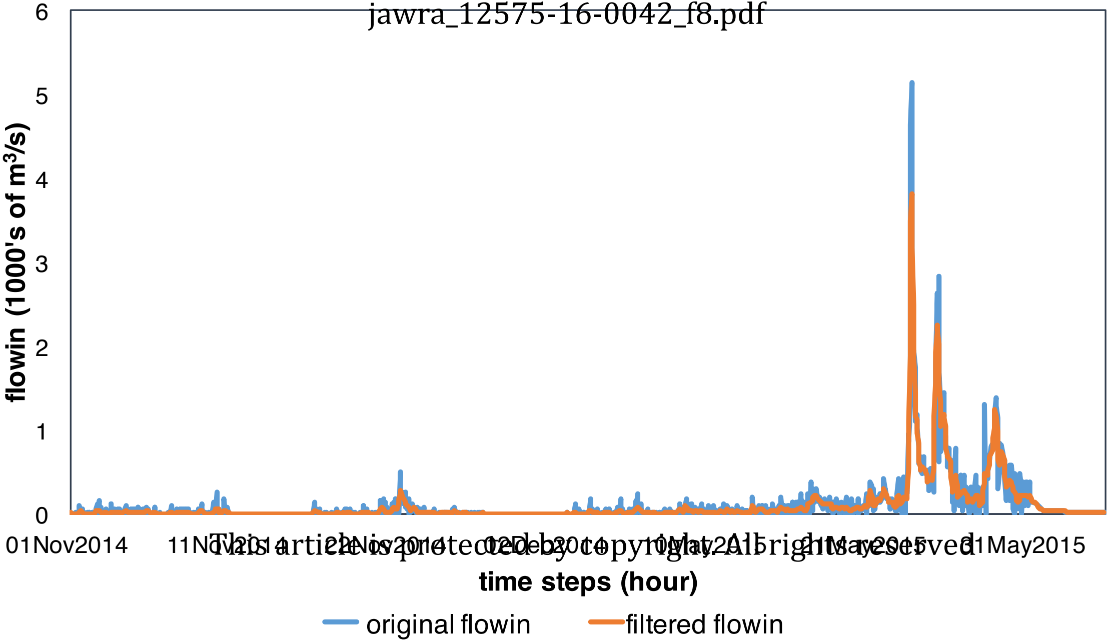





Figure 7.c. *Reservoir Inflow* at Level 3

jawra_12575-16-0042_f8.pdf



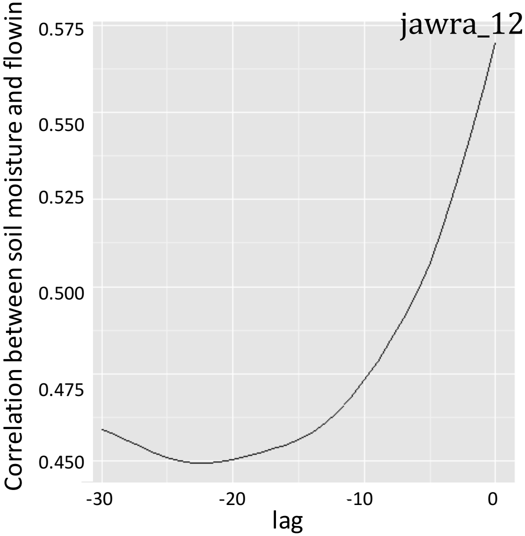



FIGURE 9.a. Correlation between Soil Moisture and *Flowin*

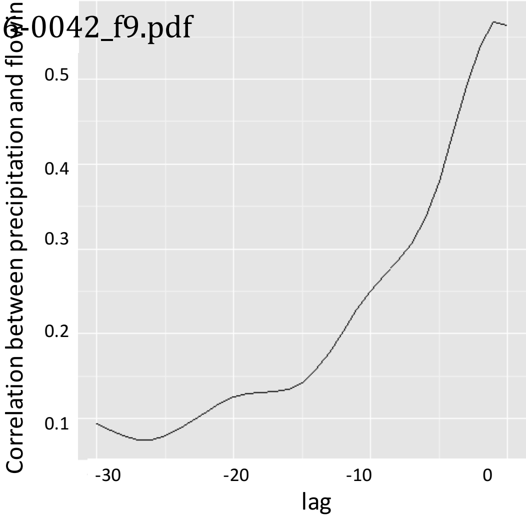



FIGURE 9.b. Correlation between Precipitation and *Flowin*

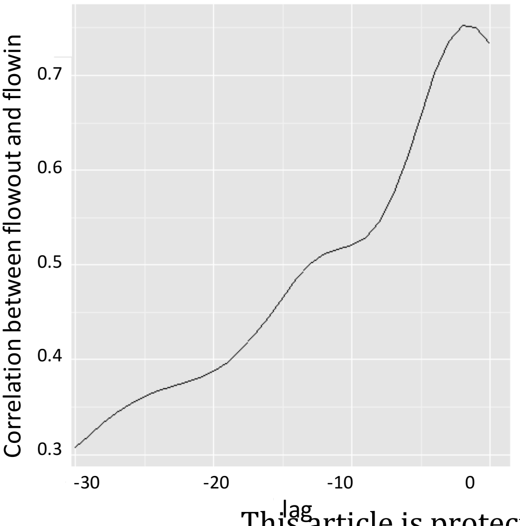
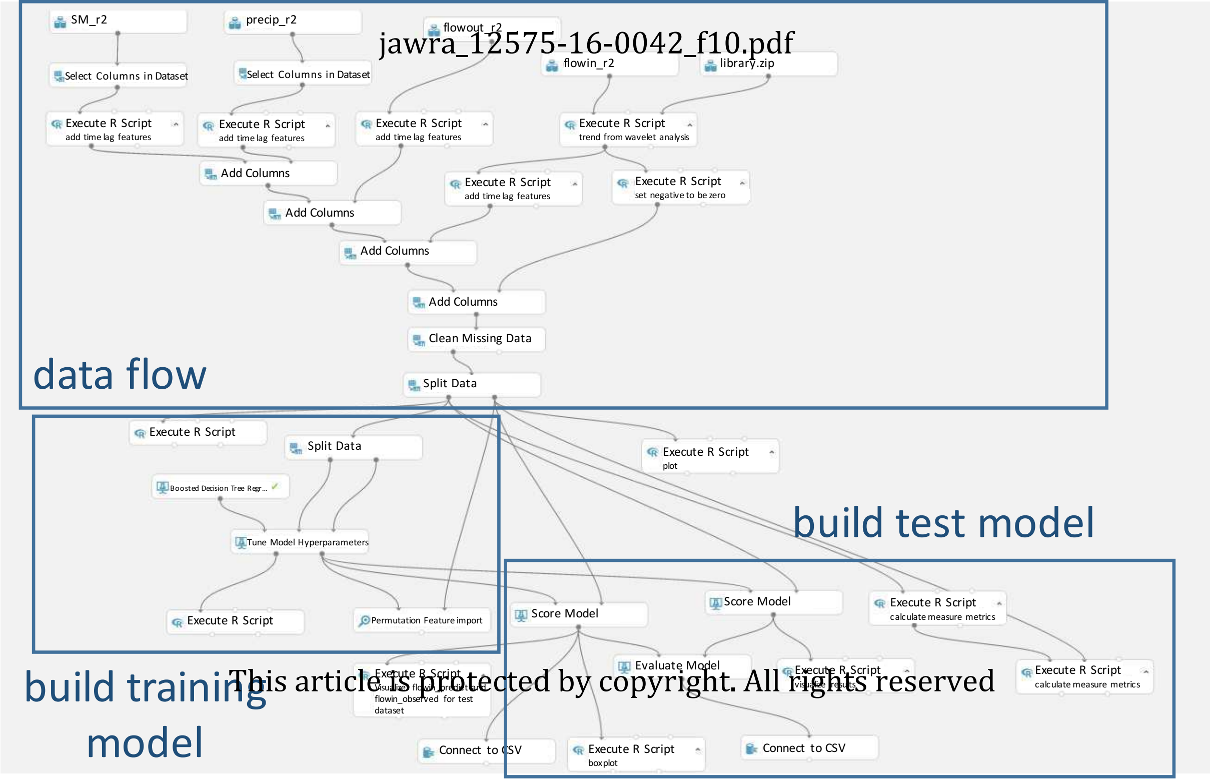
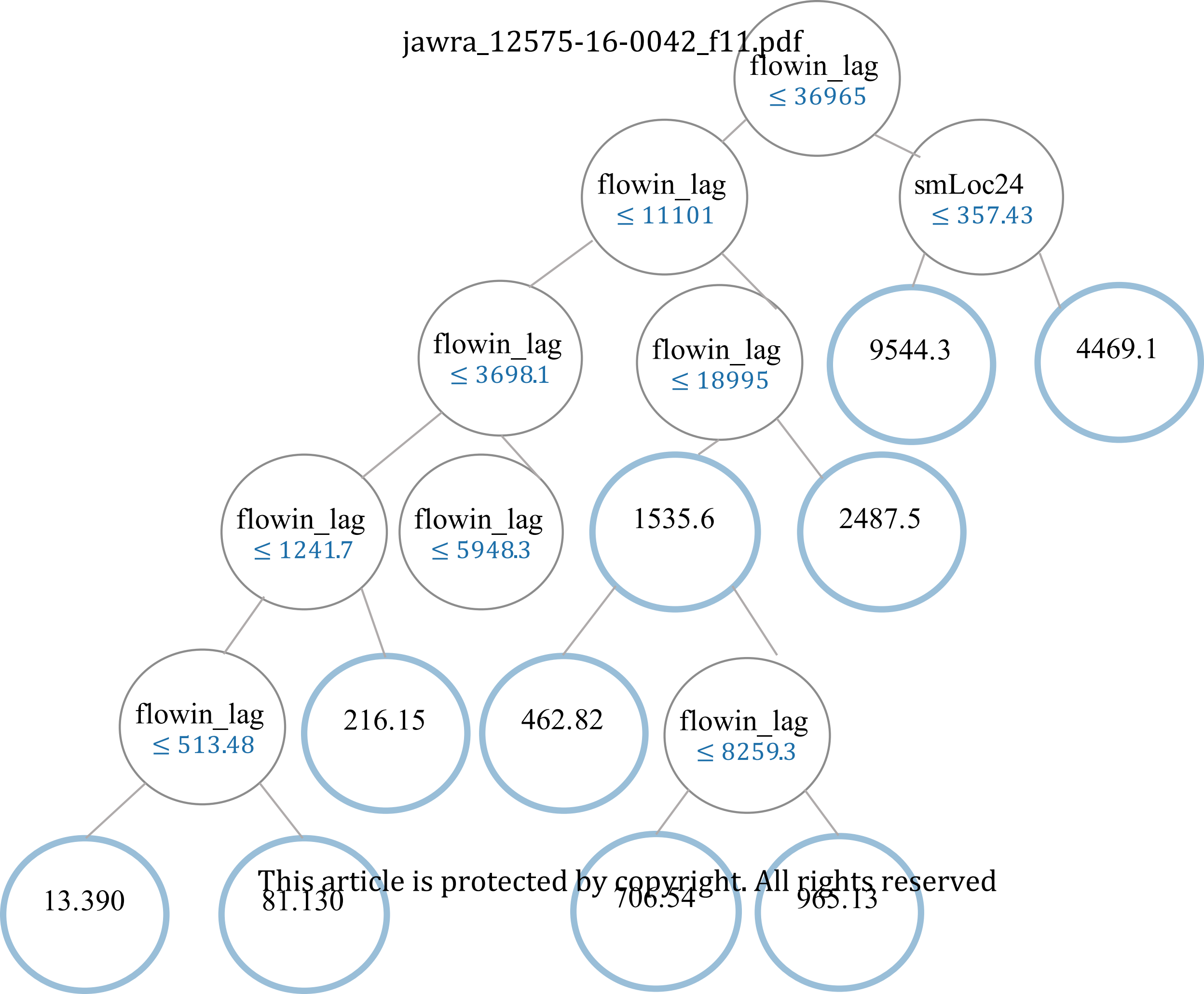


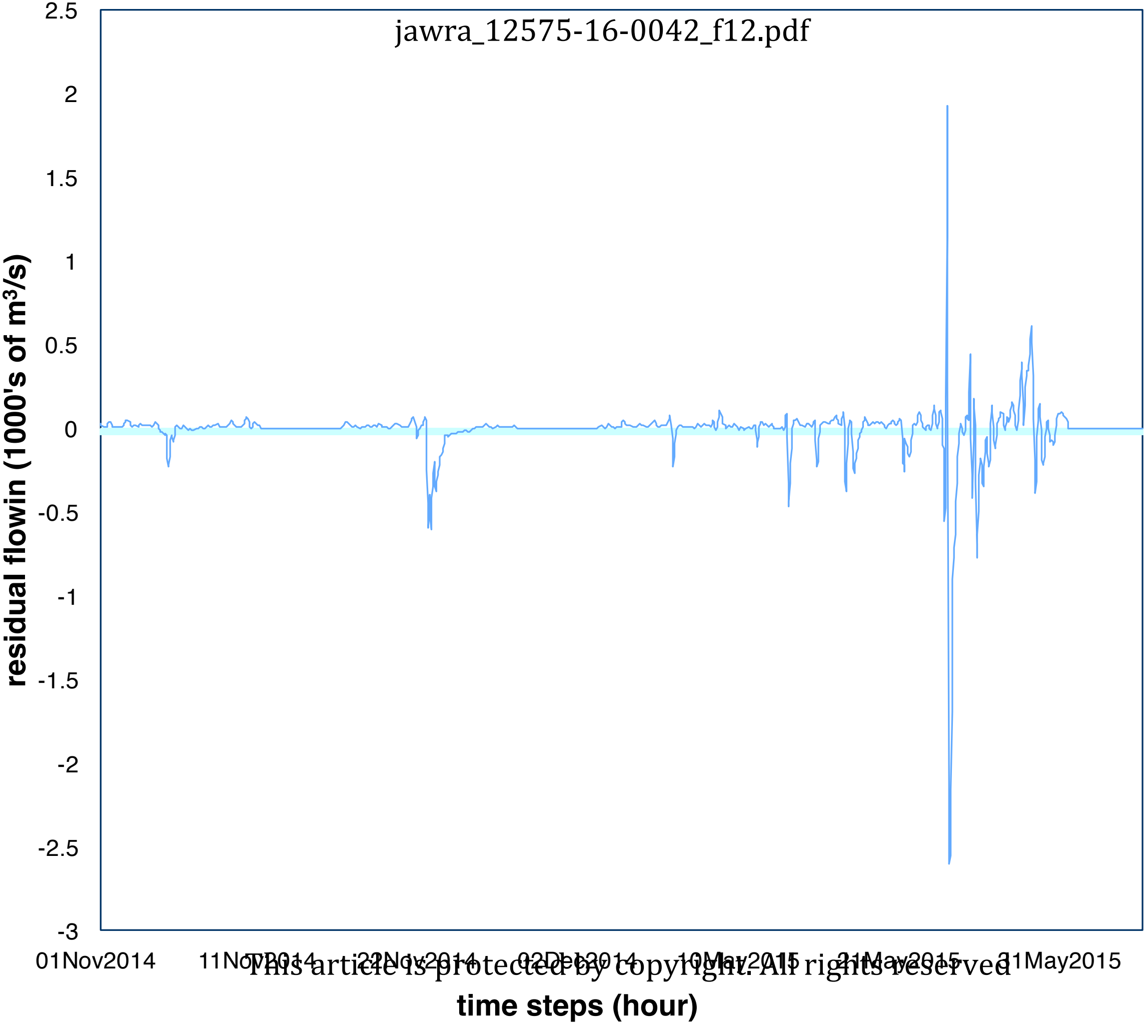



FIGURE 9.c. Correlation between *Flowout* of Upstream Reservoir and *Flowin*

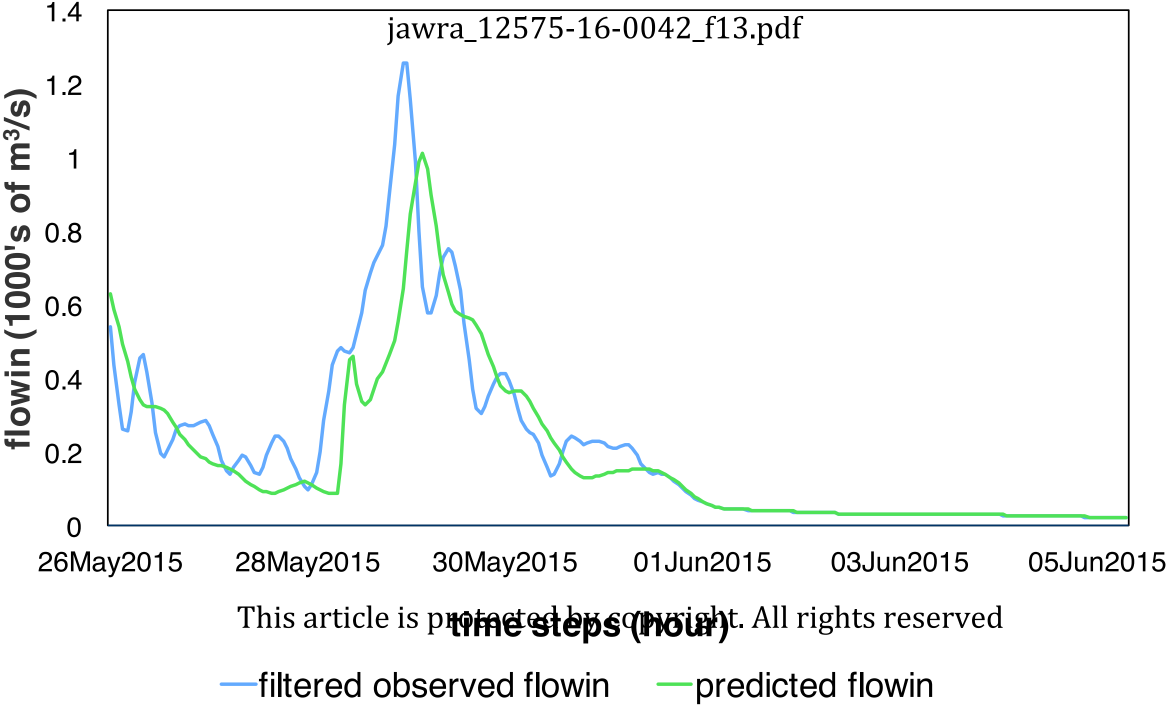

data flow
SM_r2
precip_r2
flowout_r2
flowin_r2
library.zip
Select Columns in Dataset
Select Columns in Dataset
Execute R Script
add time lag features
Execute R Script
add time lag features
Execute R Script
add time lag features
Execute R Script
trend from wavelet analysis
Add Columns
Add Columns
Execute R Script
add time lag features
Execute R Script
set negative to be zero
Add Columns
Add Columns
Clean Missing Data
Split Data
build training model
Execute R Script
Split Data
Boosted Decision Tree Regr...
Tune Model Hyperparameters
Execute R Script
Permutation Feature import
Execute R Script
plot
build test model
Score Model
Score Model
Execute R Script
calculate measure metrics
Execute R Script
Visualize flowin_predict and flowin_observed for test dataset
Evaluate Model
Execute R Script
visualize results
Execute R Script
calculate measure metrics
Connect to CSV
Execute R Script
boxplot
Connect to CSV

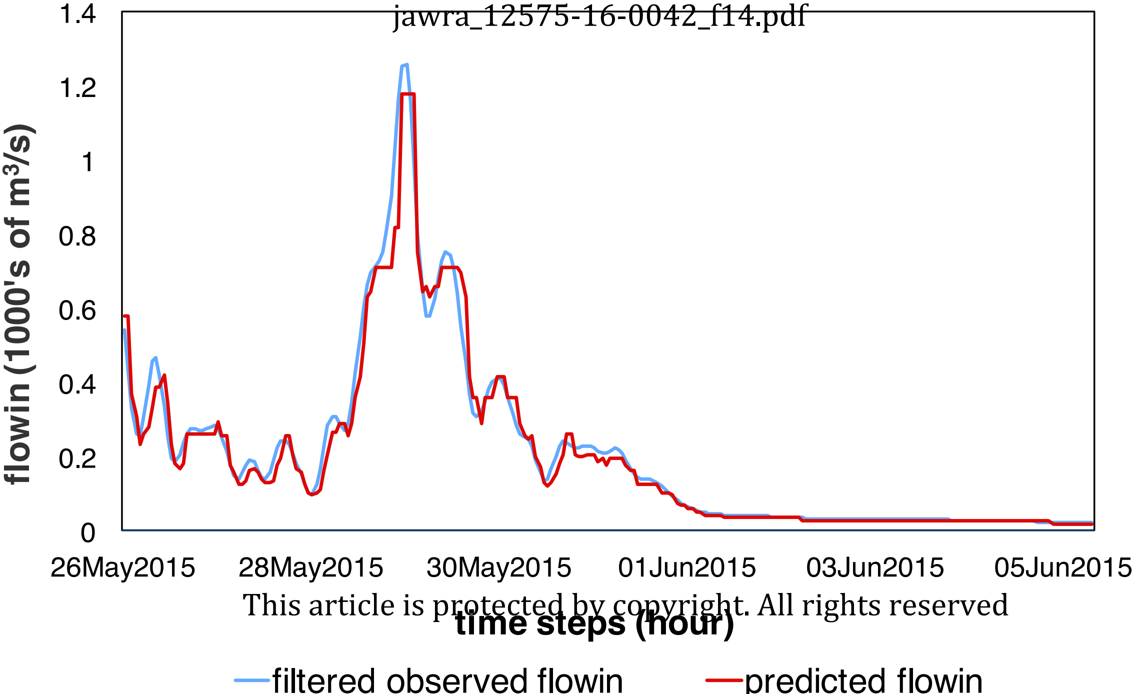

flowin_lag
≤ 36965
flowin_lag
≤ 11101
smLoc24
≤ 357.43
flowin_lag
≤ 3698.1
flowin_lag
≤ 18995
9544.3
4469.1
flowin_lag
≤ 1241.7
flowin_lag
≤ 5948.3
1535.6
2487.5
flowin_lag
≤ 513.48
216.15
462.82
flowin_lag
≤ 8259.3
13.390
81.130
706.54
965.13

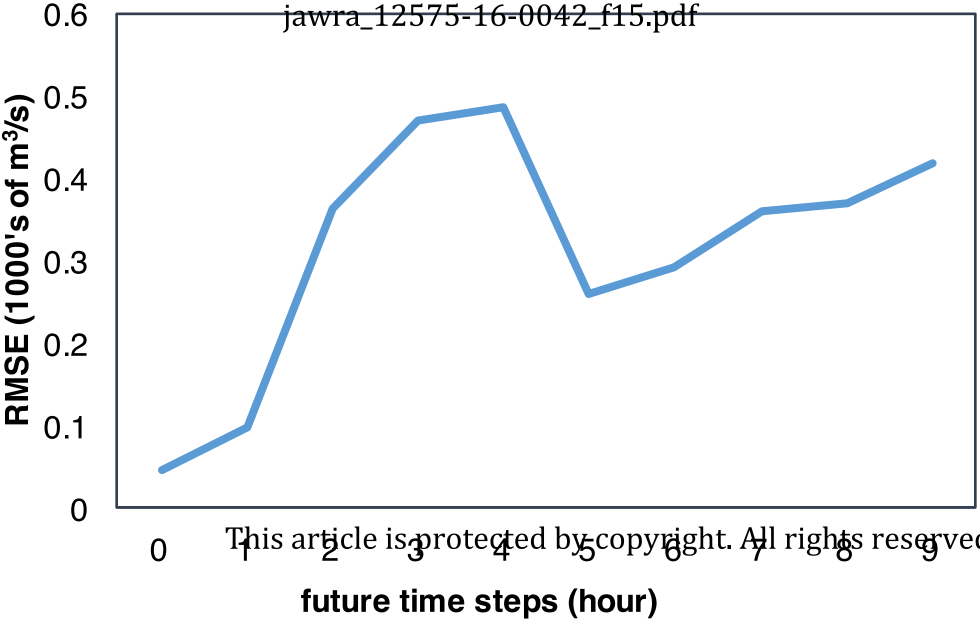
jawra_12575-16-0042_f12.pdf

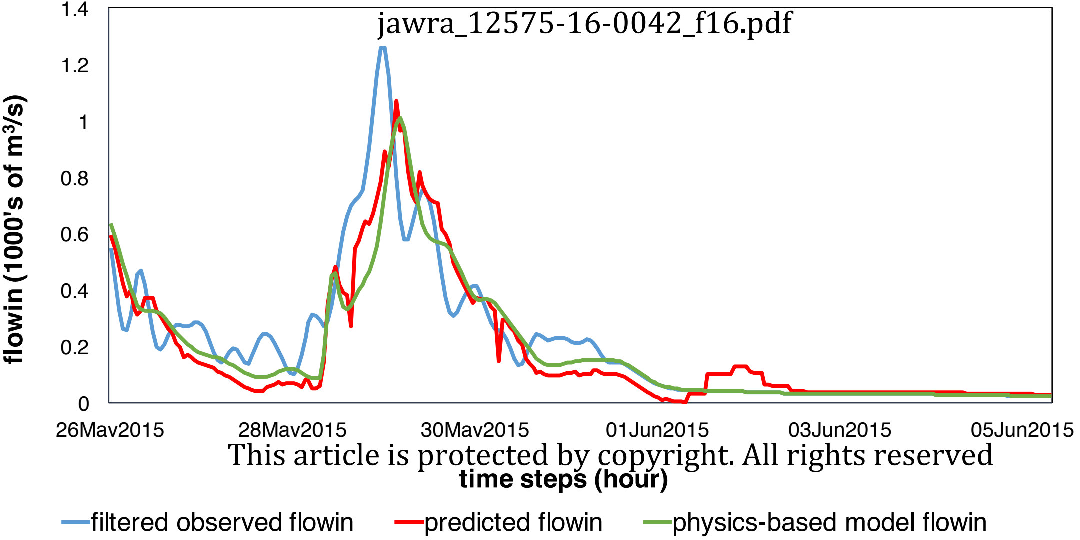


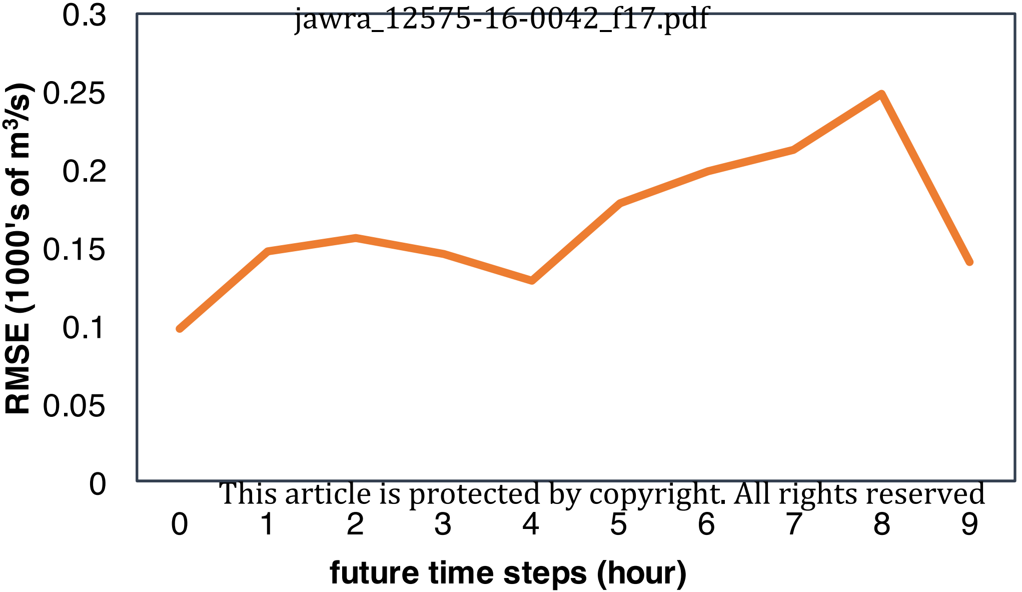
jawra_12575-16-0042_f13.pdf




1.4
1.2
1
0.8
0.6
0.4
0.2
0
flowin (1000's of m3/s)
26May2015
28May2015
30May2015
01Jun2015
03Jun2015
05Jun2015
time steps (hour)
filtered observed flowin
predicted flowin


0.6
0.5
0.4
0.3
0.2
0.1
0
RMSE (1000's of m³/s)
0
1
2
3
4
5
6
7
8
9
future time steps (hour)


1.4
1.2
1
0.8
0.6
0.4
0.2
0
flowin (1000's of m3/s)
26May2015
28May2015
30May2015
01Jun2015
03Jun2015
05Jun2015
time steps (hour)
filtered observed flowin
predicted flowin
physics-based model flowin


0.3
0.25
0.2
0.15
0.1
0.05
0
RMSE (1000's of m³/s)
0
1
2
3
4
5
6
7
8
9
future time steps (hour)

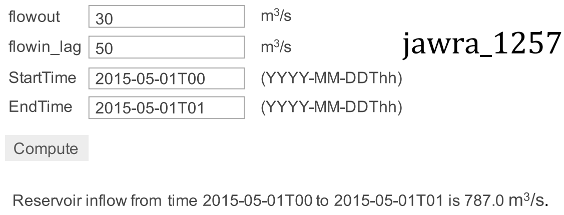



FIGURE 18.a. Flow Prediction Model to Calculate Reservoir Inflow

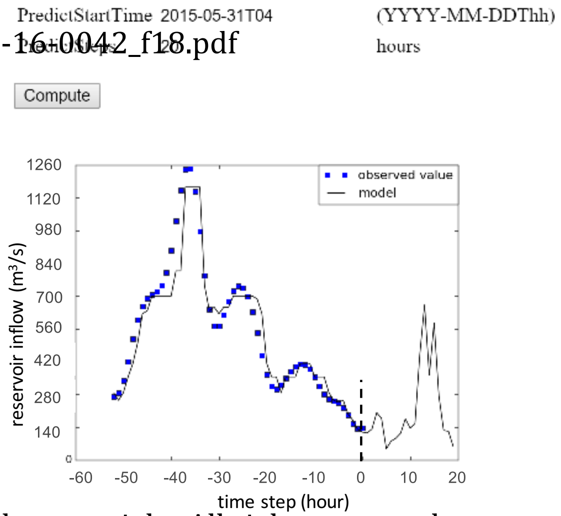
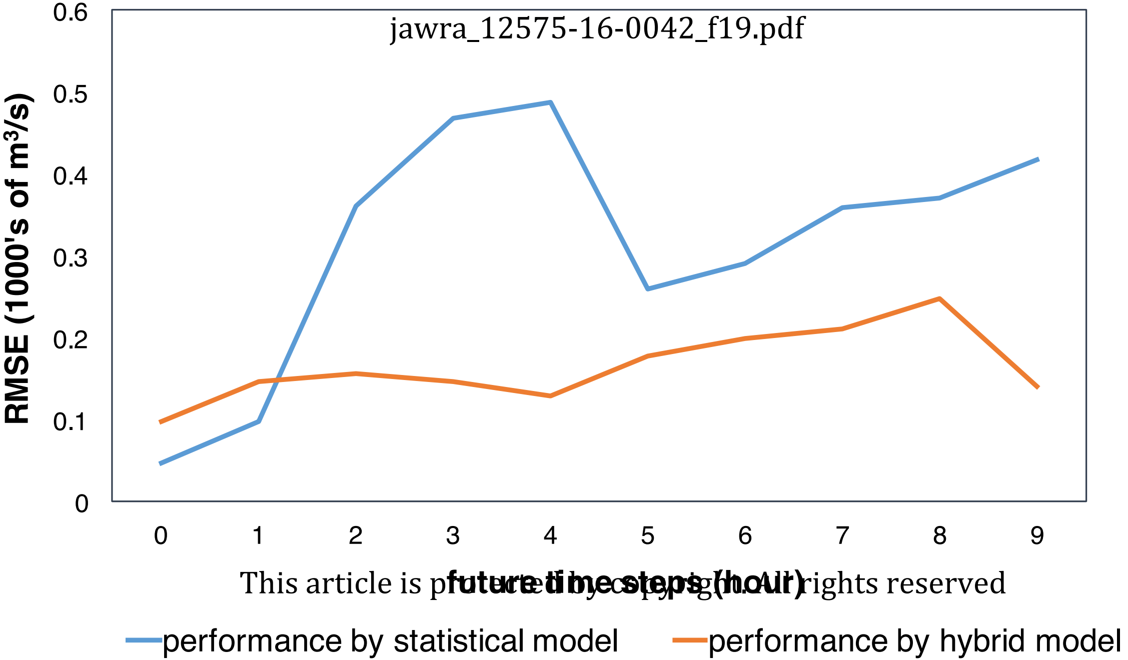



FIGURE 18.b. Reservoir Inflow Prediction


RMSE (1000's of m³/s)
0.6
0.5
0.4
0.3
0.2
0.1
0
0
1
2
3
4
5
6
7
8
9
future time steps (hour)
performance by statistical model
performance by hybrid model