https://doi.org/10.5194/hess-2020-460
Preprint. Discussion started: 7 October 2020

Hydrology and Earth System Sciences
Discussions

Modeling and interpreting hydrological responses of sustainable urban drainage systems with explainable machine learning methods

Yang Yang[1], Ting Fong May Chui[1]

[1]Department of Civil Engineering, The University of Hong Kong, Hong Kong SAR, China

Correspondence to: Ting Fong May Chui (maychui@hku.hk)

Abstract. Sustainable drainage systems (SuDS) are decentralized stormwater management practices that mimic the natural drainage processes. Their modeling is often challenged by insufficient data and unknown factors affecting the hydrological processes. This study uses machine learning methods to model directly the correlation between hydrological responses and rainfalls at fine temporal scales in two catchments of different sizes. A feature engineering method is developed to extract useful information from rainfall time series and is used in combination with a nested cross-validation procedure to derive high-quality models and to estimate their generalization errors. The SHAP method is adopted to explain the basis of each prediction, which is then used for estimating catchment response time and hydrograph separation. The explanations of the predictions provide valuable insights into the models' behavior and the involved hydrological processes. Thus, interpreting machine learning models is found as a useful way to study catchment hydrology.

1 Introduction

Sustainable drainage systems (SuDS), also known as low impact development (LID), green infrastructures (GIs), and sponge city, are decentralized stormwater management practices, which aim to promote onsite infiltration, storage, evapotranspiration, and reuse of stormwater (Fletcher et al., 2015; Jones and Macdonald, 2007). SuDS are effective in improving stormwater runoff quality, reducing runoff volume, and restoring natural hydrological regimes (Selbig et al., 2019; Trinh and Chui, 2013; Zhou, 2014). The commonly used SuDS include bioretention cells, green roofs, porous pavements, and rain barrels (Charlesworth, 2010; Gimenez-Maranges et al., 2020). It is essential to be able to predict the hydrological performance of SuDS and understand the involved hydrological processes to support their design optimization, planning, and policy making (Pappalardo and La Rosa, 2020; Yang and Chui, 2018).

A number of numerical modeling methods of various complexities have been developed for SuDS (Elliott and Trowsdale, 2007; Liu et al., 2014). The simplest methods are perhaps those developed based on the empirical equations for computing the drainage impact of different land uses in terms of peak flow or runoff volume. For instance, the rational method and the SCS runoff curve number method are modified and used in Montalto et al. (2007) and Damodaram et al. (2010) to study the effectiveness of SuDS at catchment scales. The empirical equation-based methods can be useful in preliminary designs to rapidly estimate some key performance metrics of SuDS. However, the variations in the detailed designs of SuDS can hardly

https://doi.org/10.5194/hess-2020-460
Preprint. Discussion started: 7 October 2020


Hydrology and Earth System Sciences
Discussions

be reflected in these methods. For example, it is often unclear how to vary the curve number in the SCS method when the substrate depth of green roofs changes (Fassman-Beck et al., 2016). The relatively coarse temporal and spatial resolutions of the empirical equation-based methods also limit their applications in some cases.

Process-based models are another type of methods for modeling SuDS, in which physically-based or empirical equations are used to characterize the involved hydrological processes. SuDS are typically represented in process-based models as hydrological functional units, whose properties are determined by a set of parameters. Commonly used models, including SWMM and MUSIC, are reviewed in Eckart et al. (2017) and Elliott and Trowsdale (2007). Ideally, a process-based model can be set up for a catchment using only measured values of its physical properties (Refsgaard and Storm, 1990). However, not all the parameters are measurable or can be measured at a reasonable cost. Model calibrations are often required and critical for SuDS (Platz et al., 2020). For example, in SWMM, the initial soil moisture deficit parameter of SuDS is often determined through calibration (Rosa et al., 2015). Model calibration, however, can be difficult and subject to considerable uncertainty (Schuol and Abbaspour, 2006).

The application of process-based models faces several challenges. First, it can be difficult to find a suitable model representation for SuDS. The design of SuDS can vary greatly in their installation locations, layer compositions, and materials used. Models that are capable of modeling all these design variants may not exist. Modular form models, such as GIFMod, represent SuDS as a collection of hydrological functional components arranged in 1D- or 2D-grids, allowing different design variants to be modeled (Massoudieh et al., 2017). These models, however, are arguably more complex and do not necessarily produce more accurate predictions due to higher data requirements for model setup. Second, the complex hydro-environmental processes of SuDS and the surrounding environments are difficult to model using existing models. For instance, the macropore flow in the soil layer of SuDS is not accounted for in SWMM (Niazi et al., 2017), and models for assessing the performance of SuDS in shallow groundwater environment (Zhang and Chui, 2019) and cold climate (Johannessen et al., 2017) are limited. Third, the assumptions used in process-based models may be invalid in some cases due to unknown issues related to construction, maintenance, or changes in physical properties during the service life of SuDS. For instance, the permeability of porous pavement changes over time (Yong et al., 2013), but it can be difficult to understand the processes that contribute to the clogging and the consequences. The complex environment processes or human errors, such as changes in ecological processes (Levin and Mehring, 2015) and inferior constructions, are often unknown to the modelers and difficult to predict. Therefore, other types of models that rely less on model assumptions are preferred under certain circumstances.

Machine learning methods, also referred to as data-driven modeling, predictive modeling, and statistical learning, may be used to learn the statistical correlations between interested variables of a SuDS system, and do not require the involved processes to be specified (Solomatine and Ostfeld, 2008). Machine learning methods have been widely used in many subjects in hydrological studies (Maier and Dandy, 2000), e.g., rainfall-runoff modeling (Worland et al., 2018), evapotranspiration forecasting (Karbasi, 2018), and groundwater modeling (Jang et al., 2020).

Only a few studies adopted machine learning methods to investigate the hydrological processes of SuDS. Eric et al. (2015) and Khan et al. (2013) used multiple linear regression models to predict the hydrological performance of SuDS, such as the

https://doi.org/10.5194/hess-2020-460
Preprint. Discussion started: 7 October 2020


Hydrology and Earth System Sciences
Discussions

captured runoff volume. Li et al. (2019) used neural networks to model the correlation between the characteristics of rainfall events and the runoff volumes and peak flows. Note that most of the current studies are conducted at an event level, that the response of SuDS at each time step is not modeled. Yang and Chui (2019) showed that machine learning methods, such as deep learning methods and random forest methods, are useful for predicting runoff response of SuDS at fine temporal scales, provided that the model's input variables are appropriate. However, a method to define these variables is not provided.

The lack of popularity of machine learning models in SuDS-related studies may be explained by several factors. First, the hydrological responses of SuDS are controlled by relative long-term hydrometeorological conditions. Thus, modeling the responses of SuDS at fine temporal scales requires high-dimensional hydrometeorological time series to be used as input, which is difficult in machine learning (Nielsen, 2019). In addition, machine learning methods may not have clear advantages over the equation-based methods when applied to study the performance of SuDS at an event level.

Second, machine learning models are often criticized for being lack of transparency, meaning that the basis for making each prediction is difficult to analyze (Solomatine et al., 2008). However, such information is important for model diagnosis, understanding the involved processes, and supporting decision making. To explain the basis of model prediction in hydrology, previous studies have analyzed the rules used in model trees for making runoff prediction (Solomatine and Dulal, 2003), examined the temporal evolution of hidden values of neural networks (Kratzert et al., 2019), and referred to similar events occurred in the past using instance-based learning (Wani et al., 2017). The linear regression models are also explainable as the importance of an input feature can be measured by its weight. However, as shown in Lundberg and Lee (2017), the model explanation method, SHAP, is a unified approach for explaining individual predictions while satisfying three desirable properties (details are provided in Section 2.1.3), and the other explanation methods all have some drawbacks. The SHAP method, however, has rarely been used (if any) in hydrological studies.

Third, it was not until recently, many observation data of SuDS and urban drainage systems became available (Eggimann et al., 2017), as they were relatively new technologies. The lack of data for training machine learning models naturally results in few studies applying these methods.

This study has three objectives, corresponding to the three reasons explaining the insufficient use of the machine learning methods in SuDS modeling. (1) Evaluate the usefulness of machine learning methods in predicting the hydrological responses of SuDS at fine temporal scales. (2) Explain individual predictions using SHAP, and further analyze the representation of the hydrological processes learned by the models for catchment response time estimation and hydrograph separation. (3) Develop and present tools and methods for building higher quality machine learning models for SuDS-related studies and demonstrate the applications.

https://doi.org/10.5194/hess-2020-460
Preprint. Discussion started: 7 October 2020


Hydrology and Earth System Sciences
Discussions

2 Methods and material

2.1 Methods

Let Y_t denote the hydrological response of SuDS at time t, and $X_{t-m,t}$ denote its hydro-environmental conditions measured between time $t-m$ and t. Y_t can be a numerical or categorical random variable, e.g., the outflow rate or the binary observation of the dry/wet condition of the underdrain system of SuDS. $X_{t-m,t}$ is a random vector, which can be written as $(P_{t-m}, P_{t-m+1}, \cdots, P_{t-0}, E_1, E_2, \cdots, E_n)^T$, where P_{t-i} is the rainfall depth measurement at time $t-i$, E_i is the ith hydro-environmental variable measured between time $t-m$ and t, and T is the transpose operator. Each element is termed as a feature.

It is assumed that the relationship between Y_t and $X_{t-m,t}$ can be described by some unknown function f, as shown in Eq. (1),

$$Y_t = f\left(X_{t-m,t}\right) + \epsilon\,, \tag{1}$$

where, ϵ is a random error term.

The task of machine learning is to find $\hat{f}$, the estimate of f, based on observed data of $X_{t-m,t}$ and Y_t, such that $\widehat{Y_t}$ is an accurate prediction for Y_t (Eq. (2)),

$$\widehat{Y_t} = \hat{f}\left(X_{t-m,t}\right) \tag{2}$$

An $\hat{f}$ can be derived by fitting a machine learning method to the observed data of $X_{t-m,t}$ and Y_t, $\left\{\left(x_{t-m,t,1}, y_{t,1}\right), \left(x_{t-m,t,2}, y_{t,2}\right), \cdots, \left(x_{t-m,t,n}, y_{t,n}\right)\right\}$.

2.1.1 Nested cross-validation for feature engineering, hyperparameter optimization, model selection, and generalization error estimation

The runoff generation processes of SuDS are usually regulated by the antecedent soil moisture, which may be controlled by the hydro-environmental factors of the past few days or weeks. However, as the number of dimensions of $X_{t-m,t}$ increases, the amount of data required for model training may also increase rapidly for the fitted models to generalize properly to unseen data (Verleysen and François, 2005). To avoid training models directly on high-dimensional data, $X_{t-m,t}$ can be transformed to lower-dimensional features through some function φ, and then $\hat{g}$, the function that maps the features to $\widehat{Y_t}$, is estimated (Eq. (3)).

$$\widehat{Y_t} = \hat{g}\left(\varphi(X_{t-m,t})\right) \tag{3}$$

The form of φ is arbitrary, and the process of defining φ is called feature engineering (Kuhn and Johnson, 2019). Summary statistics of the input variable, such as the mean and the standard deviation, are commonly used as its features. Other commonly used methods include principal component analysis and wavelet analysis. Nielsen (2019) reviewed the feature engineering

https://doi.org/10.5194/hess-2020-460
Preprint. Discussion started: 7 October 2020


Hydrology and Earth System Sciences
Discussions

methods for time series data and showed that domain knowledge can be applied to generate useful features to solve problems involving financial or healthcare time series.

In this study, φ is defined based on the knowledge of the hydrological processes of SuDS. SuDS generally have fast responses to rainfall due to their small sizes and the presence of overland flow and shallow interflow (DeBusk et al., 2011). Thus, Y_t is more affected by the recent hydro-environmental conditions than by that of the past, especially when Y_t is a variable that is related to runoff. Therefore, it is possible to compress the high-dimensional rainfall time series $(P_{t-m}, P_{t-m+1}, \cdots, P_{t-0})^T$ into a shorter representation by discarding information that is less relevant to predicting Y_t. This study assumes that the temporal distribution of rainfalls of the relatively distant past is of little relevance to Y_t while their accumulated depths are potentially important. The aggregated rainfall depths over different time periods, with an emphasis on compressing rainfalls of the distant past, are derived and used as input to machine learning models.

$D_{t-a,t-b}$, the aggregated rainfall depth between time $t-b$ and $t-a$, is calculated using Eq. (4),

$$D_{t-a,t-b} = \sum_{i=a}^{b} P_{t-i}, \tag{4}$$

where, a and b are integers, and $0 \le a \le b$. a and b are respectively chosen from set S_a and set S_b, as defined in Eq. (5) and Eq. (6),

$$S_a = \{x \in \mathbb{N} | 0 \le x \le l\} \cup \{x \in \mathbb{N} | (\exists k \in \mathbb{N})[x = f(k) + l + 1 \text{ and } x \le m]\}, \tag{5}$$

$$S_b = \{x \in \mathbb{N} | 0 \le x \le l \text{ or } x = m\} \cup \{x \in \mathbb{N} | (\exists k \in \mathbb{N})[x = f(k) + l \text{ and } x \le m]\}, \tag{6}$$

where, $l, m \in \mathbb{N}^+$ and $l \le m$, and

$$f(k) = \sum_{i=0}^{k} u_i, \tag{7}$$

where, $u_0 = 0$, $u_1 = 2$, $u_i = \text{round}(2 * q^{i-1})$ for $i \ge 2$, and $q > 1$.

Computing $D_{t-a,t-b}$ using the elements of S_a and S_b may be interpreted as placing a set of cut points along the time axis and computing accumulated rainfall depths between these points, as shown in Fig. 1a. The largest element in S_a and the second largest element S_b may be removed if they correspond to a cut point that is closer to $t-m$ than to the other neighboring cut points, because the intervals form between two neighboring cut points should not become smaller when scanning the time axis from t through $t-m$.

The elements of S_a and S_b are determined by m, l, and q. Intuitively, l specifies a period between $t-l$ and t, the rainfalls recorded during which are considered to be recent and important for predicting Y_t. m specifies a lookback period, that rainfall time series recorded outside the period between $t-m$ and t are regarded as irrelevant to Y_t. q corresponds the importance of a more recent rainfall record for predicting Y_t when compared to a record in the proceeding interval (formed by two neighboring cut points, as shown in Fig. 1a). The length of the intervals proceeding $t-l$ roughly formed a geometry sequence with the common ratio q.

https://doi.org/10.5194/hess-2020-460
Preprint. Discussion started: 7 October 2020


Hydrology and Earth System Sciences
Discussions

There are several options to select (a, b) pairs from S_a and S_b for calculating the rainfall depth features. (1) a takes each value in S_a, and b takes the smallest value in S_b that satisfies $a \leq b$. (2) $a = 0$, and b takes each value in S_b. (3) b takes each value in S_b, and $a = b$ when $b \leq l$ and $a = 0$ when $b > l$. (4) Similar to (3), except $a = l + 1$ when $b > l$. An example is given in Fig. 1a. The (a, b) pairs may also be drawn randomly from S_a and S_b, as long as $a \leq b$ is satisfied.

The collection of $D_{t-a,t-b}$ are compact representation of the original rainfall time series. The resolution and the encoded information of this representation are controlled by m, l, and q. An example is given in Fig. 1b, where the number features decreased rapidly when q increases. However, m, l, and q must be set before training machine learning models. Thus, they are termed as feature engineering "hyperparameters" instead of "parameters", which can be learned during training. The term, "hyperparameters" also appear in many machine learning methods. They control the model complexity and learning behaviors, e.g., the number of neighbors in K-Nearest neighbor method and the number of trees in random forests method (Kuhn and Johnson, 2013).

The effectiveness of a set of hyperparameters can be assessed by evaluating the prediction errors of the resulted model on test datasets. The k-fold cross-validation (CV) approach is often used to estimate prediction errors when observed data is limited (Zheng et al., 2018). In CV, the observed data are randomly split into k folds, and for each iteration, $k - 1$ folds are used for fitting machine learning models and the remaining fold (i.e., the validation fold) is used to assess their prediction errors, until all folds have been used for model evaluation once. The CV approach can also be used for choosing between models trained using different machine learning methods. However, it should be noted that prediction errors obtained for the validation folds should not be interpreted as the errors expected for unseen data, i.e., the generalization error (Cawley and Talbot, 2010).

This study adopts a nested CV scheme for hyperparameter optimization and generalization error estimation, as shown in Fig. 1c. The data is first split into multiple sets of training (outer) and test folds. At each outer CV iteration, a training (outer) fold is further split into multiple training (inner) and validation folds, which are used in the inner CV. At inner CV iterations, different machine learning methods with different sets of hyperparameters are trained on the training (inner) folds and tested on the corresponding validation folds. The machine learning method and hyperparameters that minimize the average inner CV errors are then selected and fitted to the training (outer) fold. The fitted model is then evaluated using the test fold of outer CV iteration. Thus, at each outer CV iteration, the optimal machine learning method and hyperparameters are identified, and an outer CV error and a set of inner CV errors are obtained. The outer CV error can be considered as an estimation of the generalization error.

It should be noted that the optimal method and hyperparameters may vary at different outer CV iterations. The outer CV error reflects the effectiveness of the entire process for deriving a machine learning model using a given dataset (i.e., the outer training fold), which includes the selection of candidate machine learning methods and hyperparameters sets, the feature engineering methods, and the inner CV procedures for model comparison and selection. Thus, the specific optimal methods and hyperparameters identified at different outer CV iterations are not very important. These configurations need to be re-estimated by treating the entire dataset as a training (inner) set and applying the aforementioned process for deriving a machine learning model.

https://doi.org/10.5194/hess-2020-460
Preprint. Discussion started: 7 October 2020


Hydrology and Earth System Sciences
Discussions

2.1.2 Tree boosting methods and XGBoost

Tree boosting methods presented in Friedman (2001) is used in this study to build machine learning models. The XGBoost machine learning software is employed for model training for its improved regularization methods, high computational efficiency, and ability to achieve state-of-the-art results on many machine learning tasks (Chen and Guestrin, 2016; Chen and He, 2020; Nielsen, 2016). Tree boosting methods build multiple regression trees that the final prediction of an input is the sum of the predictions of individual trees, as shown in Eq. (8) (Chen and Guestrin, 2016),

$$\hat{y}_i = \theta(x_i) = \sum_{k=1}^{K} f_k(x_i), f_k \in \mathcal{F}, \quad (8)$$

where, K is the number of regression trees, also known as the number of boosting iterations, and $\mathcal{F} = \{f(X) = w_{q(X)}\}$ $(q: \mathbb{R}^m \to \{1,2,\dots,T\}, w \in \mathbb{R}^T)$ is the functional space of regression trees. T is the number of the leaves of the tree. w_i is the score of the ith leaf, $w_i \in \mathbb{R}$. q represents a tree structure that maps the m-dimensional input variable X to a leaf index. $\emptyset(X_{t-m,t})$ is used as the input variable in this study.

The trees are built to minimize the objective function in Eq. (9),

$$\mathcal{L}(\theta) = L(\theta) + \Omega(\theta), \quad (9)$$

where L is the training loss; $L(\theta) = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2$, where n is the number of observations in the training dataset. Ω is the regularization term that penalizes complex models. $\Omega(\theta) = \sum_{k=1}^{K} \omega(f_k)$, and $\omega(f_k) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2$. γ and λ are hyperparameters that need to be set before training. The regression trees are built in a stage-wise manner, that one new tree is added to the ensemble of trees built in pervious steps at a time. The methods used in XGBoost to derive the tree structures can be found in Chen and Guestrin (2016), and Mitchell and Frank (2017).

A number of hyperparameters, besides γ and λ, are used in XGBoost to regulate the tree structure and the learning behavior. A complete list of the hyperparameters can be found in the documentation of the XGBoost software (Chen and He, 2020). The optimal set of hyperparameters can be identified by evaluating and comparing the CV errors of a number of candidate sets or by applying various optimization methods, such as Bayesian optimization (Xia et al., 2017). The XGBoost hyperparameters and the feature engineering hyperparameters are optimized together in this study through grid search, where the inner CV errors of every considered candidate set are compared.

2.1.3 Interpreting model predictions using SHapley Additive exPlanations (SHAP)

The SHapley Additive exPlanations (SHAP) method proposed by Lundberg and Lee (2017) is used in this study to explain the of basis of individual predictions, i.e., the contribution of each feature to each prediction $f(x)$, where f is a machine learning model and x is a sample of random variable X, which has M features. The attributions derived for each feature of x are termed as SHAP values. The SHAP value is related to the Shapley value in cooperative games in game theory, which is defined as the average marginal contribution of a player across all coalitions that the player is a member of (Biecek and Burzykowski, 2020;

https://doi.org/10.5194/hess-2020-460
Preprint. Discussion started: 7 October 2020


Hydrology and Earth System Sciences
Discussions

Serrano, 2007). In machine learning, a feature's SHAP value for a particular prediction describes its contribution to that prediction compared to the average prediction for the dataset (Molnar, 2020).

The SHAP value is the only local feature attribution measurement that is locally accurate, consistent, and compliant with the missingness requirement, among all the measurements defined in the family of additive feature attribution methods (Lundberg et al., 2018). The local accuracy property refers to that the sum of the feature attributions equals $f(x)$. The consistency property states that the attribution of a feature does not decrease when this feature has the same or a larger impact on the prediction when the model changes. The missingness property requires the features that have no effects on $f(x)$ to receive an attribution of 0 (Lundberg et al., 2018). The SHAP value of feature i for sample x and model f, $\emptyset_i(f,x)$, can be computed using Eq. (10),

$$\emptyset_i(f,x) = \sum_{R\in\mathcal{R}} \frac{1}{M!}[f_x(P_i^R \cup i) - f_x(P_i^R)], \quad (10)$$

where $f_x(S) = E[f(X)|\mathrm{do}(X_s = x_s)]$, S is a set of features of random variable X, and x_s is a vector containing the entries of x corresponding to the features in S(Lundberg et al., 2020). R is a random ordering of all the features, and $\mathcal{R}$ is the space of all possible orderings. M is the number of features of X. P_i^R is the set of all features that come before i in ordering R.

The SHAP values, however, can be extremely computationally expensive to calculate (Lundberg et al., 2018). The TreeExplainer algorithms developed in Lundberg et al. (2020) are efficient in calculating the SHAP values for tree-based models. TreeExplainer is designed such that the unique structures of trees are utilized to simultaneously account for all possible subset S, allowing the SHAP values to be computed in polynomial time. More details on TreeExplainer can be found in Lundberg et al. (2020).

TreeExplainer is employed in this study to calculate the SHAP values for the XGBoost models to assess the contribution of the rainfall depth features and the other hydro-environmental features to each prediction. The SHAP values reflect the contributions of the rainfalls of different periods to the runoff at the current time step, which is frequently studied in rainfall-runoff modeling (Beven, 2012). In this study, the temporal dynamics of the rainfall-runoff relationship are visualized using hydrographs that present the SHAP value composition of the predicted flow rate at each time step (thanks to the local accuracy property).

The contribution of rainfall depth recorded at time $t-k$ to a prediction $f(x)$, $\tau_{P_{t-k}}(f,x)$, can be calculated using Eq. (11),

$$\tau_{P_{t-k}}(f,x) = \sum_{(i,j)\in S|i\le k\le j} \emptyset_{D_{t-i,t-j}}(f,x)\frac{p_{t-k}}{d_{t-i,t-j}}, \quad (11)$$

where, S is the set of (i,j) that is used for calculating rainfall depth features $D_{t-i,t-j}$ used in f, $\emptyset_{D_{t-i,t-j}}(f,x)$ is the SHAP value for $D_{t-i,t-j}$. p_{t-k} is the observed value of P_{t-k}. $d_{t-i,t-j}$ is the observed value of $D_{t-i,t-j}$. Eq. (11) essentially means that $\emptyset_{D_{t-i,t-j}}(f,x)$ is distributed proportionally among P_{t-k} according to their values. $\tau_{P_{t-k}}(f,x)$ is useful for understanding the continuing impact of a rainfall depth record on the runoff or further attributing the runoff to rainfall of each time step.

The feature engineering hyperparameters, such as m, l, and q, carry physical meaning (as defined in Section 2.2.1 and Table 1), it might be useful to investigate how their values affect the inner CV error of the resulted models. Thus, an XGBoost model

https://doi.org/10.5194/hess-2020-460
Preprint. Discussion started: 7 October 2020


Hydrology and Earth System Sciences
Discussions

is built in this study to predict the inner CV error based on the feature engineering hyperparameters. The attributions of the hyperparameters to the CV errors are then calculated using the SHAP method. The results can be useful for finding new sets of hyperparameters of higher qualities and identifying the physical factors that are related to the runoff generation processes, e.g., the duration of time over which a rainfall event affects the runoff generation process.

2.1.4 Comparing with other machine learning models and process-based models

The XGBoost models built in this study are compared with models that have simpler structures, i.e., linear regression models. This is to confirm that more sophisticated models have reasonably good performance and can outperform some "baseline" models (Chollet and Allaire, 2018). The linear models are built though a nested CV procedure, where the CV folds and the candidate hyperparameters are the same as that of the XGBoost models.

The Storm Water Management Model (SWMM) is a widely used process-based hydrological/hydraulic model for modeling stormwater runoffs in urban catchments (Rossman, 2015). The SWMM model developed in Lee et al. (2018a) is compared to machine learning models built for the same catchment in this study. In SWMM, a catchment is divided into multiple subcatchments connected by drainage networks, and SuDS are modeled as a combination of hydrological functional layers (Rossman and Huber, 2016).

2.2 Case studies

Two urban catchments of different sizes, data availabilities, and types of SuDS are investigated in this study. It is challenging to build process-based models for either site due to the lack of information on the physical properties of the catchment. The same machine learning methods are applied to both sites to show their applicability. Results obtained for site 1 are analyzed in detail in the results sections. The results of site 2 are presented in Section 3.4, where the performances of different types of models are compared.

2.2.1 Washington Street SuDS site (WS)

Site 1 locates in Washington Street, Geauga County, Ohio, U.S., where multiple types of SuDS were built to treat stormwater runoffs generated by a commercial building and the nearby parking lot, as shown in Fig. 2a (Darner et al., 2015). Runoffs from approximately half of the roof of the commercial building, i.e., an impervious area of 316 m^2, were drained to a rain garden with a surface area of 37 m^2. The 762 m^2 parking lot was constructed using porous pavements to permit infiltration. Site 1 is referred to as "WS" hereinafter.

The meteorological and hydrological conditions of WS were monitored by the U.S. Geological Survey (USGS) since 2009 (Darner et al., 2015; Darner and Dumouchelle, 2011). The outflows from the SuDS were collected by three flumes, inside which the water levels were measured by the attached pressure transducers. The water level measurements were then converted to flow rates using stage-discharge rating curves. Flumes 1, 2, and 3 respectively measured the flow rate of the surface runoff

https://doi.org/10.5194/hess-2020-460
Preprint. Discussion started: 7 October 2020


Hydrology and Earth System Sciences
Discussions

from the parking lot, the overflow from the surface layer of the rain garden, and the underdrain flow from the parking lot (Fig. 2a). Rainfalls were monitored using an onsite weather station.

The hydrological performances of SuDS in WS were investigated in Darner and Dumouchelle (2011) and Darner et al. (2015). Building process-based models for WS can be challenging because of the uncertainties of the hydrological processes, i.e., the rain garden was not isolated from the gravel storage layer of the porous pavements, permitting an unknown amount of stormwater received by the rain garden to enter the porous pavements' underdrain system. Thus, machine learning methods may be useful in this case.

In WS, the rainfall depths were recorded at 10-minute intervals. The water levels were measured every minute and were recorded at least every 10 minutes and when the water levels changed. Figure 2b shows the availability of the rainfall-runoff data recorded at the regular 10-minute intervals between 2010 to 2013 for the warm season, i.e., from April to October (Darner et al., 2015). Runoffs were not detected in flume 1 throughout the entire period. The gaps in the rainfall-runoff records in Fig. 2b correspond to missing values caused by technical issues or no record for the periods without runoffs. However, the cause of each missing value was unknown in this study. The rainfall-runoff data recorded before the large gap in rainfall data in July 2013 are used for machine learning, for which the few missing rainfall depth values are filled with 0. The total runoff of WS is computed as the sum of the runoffs of flumes 2 and 3, where the missing runoff values of flume 2 are filled with 0 (as overflows from the rain garden occurred infrequently). The total runoff observations are considered missing if the observations for flume 3 were unavailable.

In the rainfall-runoff models built for WS, the output variable Y_t is the flow rate of the total runoff measured at the regular 10-minute intervals. The input variable $\varphi(X_{t-m,t})$ is a vector of the rainfall depth features calculated using Eq. (4), and optionally the cumulative rainfall depth recorded since the beginning of the onsite monitoring and the binary vectors indicating which month the rainfall event was in (using one-hot encoding). The cumulative rainfall depth feature is associated with the long-term runoff load of SuDS, which can be useful information for predicting their hydrological properties, as they may evolve over time (Yong et al., 2013). The seasonality feature is useful if the hydrological properties of SuDS varied in different seasons. $account\_CumRain$ and $account\_season$ control whether to include the optional features or not. The feature engineering hyperparameters and their possible values are listed in Table 1. Note there are four options to define (a, b) pairs for generating the rainfall depth features (Section 2.1.1). Thus, the model training methods presented below are applied four times, each corresponded to an option.

The nested CV procedure is used to optimize the hyperparameters. The number of both the outer and inner CV folds is set to 5. Independent rainfall events are identified using a 24-hour dry spell (Guo and Senior, 2006). The CV folds are created such that (1) the data of each independent rainfall event are grouped together and allocated in the same fold, and (2) the peak flows of the rainfall events in the training (inner), validation, and test folds follow roughly the same distribution (i.e., the stratified sampling method as presented in Zeng and Martinez (2000)). This is to ensure that there are no data leakage and the data in each fold are representative. 100 sets of feature engineering hyperparameters are first randomly generated. The inner CV errors of the models resulted from each unique combination of the feature engineering hyperparameters and the XGBoost

https://doi.org/10.5194/hess-2020-460
Preprint. Discussion started: 7 October 2020


Hydrology and Earth System Sciences
Discussions

hyperparameters are then evaluated. The optimal combinations are those resulted in models with the lowest mean inner CV error, i.e., the "best models". Alternatively, if minimizing the prediction errors on different validation folds is considered as a multi-objective optimization problem (Sean, 2013), an ensemble of optimal XGBoost models may be found on the Pareto front using the fast non-dominated sorting method (Deb et al. 2002).

The rainfall-runoff relationships learned by the models are interpreted using SHAP in Section 3.3. The XGBoost models are compared to the linear regression models in Section 3.4.

2.2.2 Shayler Crossing Watershed (SHC)

The second study site is the Shayler Crossing Watershed (SHC) located east of Cincinnati, OH, U.S., as shown in Fig. 2c. SHC has an area of around 1 km^2, and its land-use type is primarily residential. The drainage system of the SHC consists of conduits and channels, detention ponds, dry ponds, and wet ponds (Lee et al., 2018a). In SHC, the stormwater runoffs generated by the indirectly connected impervious areas (such as the sidewalks) are treated by the nearby pervious areas, which are termed as buffering pervious areas (BPA) and function as SuDS (Lee et al., 2018b). An SWMM model was built in Lee et al. (2018a), where SHC was divided into 191 subcatchments, and the BPA were modeled as vegetated swales. A considerable amount of data on the physical properties of the catchment is required to set up such a spatially refined model. The model was calibrated in Lee et al. (2018a) using 2 months of 10-minute runoff data recorded from July to August 2009. In this study, the 10-minute rainfall-runoff data recorded in July 2009 is used for training machine learning model, and the data of August 2009 is used for testing. The candidate hyperparameters are the same as that of WS, except that $account\_CumRain$ and $account\_season$ are set to 0. The nested CV scheme is not used due to the small size of the dataset. The optimal set of hyperparameters are simply identified by using a validation fold derived from the July 2009 data. The XGBoost models is compared with the linear models and SWMM model in Section 3.4.

3 Results and discussion

3.1 Hyperparameter optimization and model training

In this study, the inner CV errors of models resulted from different combinations of feature engineering and XGBoost hyperparameters (Table 1) are compared to identify the optimal sets of hyperparameters at each outer CV iteration. As an example, the model training processes for three sets of hyperparameters in an outer CV iteration for WS are shown in Fig. 3a, where the feature engineering hyperparameters are fixed and the XGBoost hyperparameters vary in different sets. The solid lines correspond to the mean of the prediction errors estimated in the 5 inner CV iterations, and the width of the ribbons correspond to their standard deviations. The model's prediction accuracy on both the training folds (inner) and the validation folds initially improved rapidly with the increased number of boosting iterations for all hyperparameter sets. However, as more boosting iterations were used, the model's validation errors started to increase for hyperparameter sets 2 and 3, which is a clear indication of overfitting. The optimal number of boosting iterations of each set of hyperparameters was that corresponded to

https://doi.org/10.5194/hess-2020-460
Preprint. Discussion started: 7 October 2020


Hydrology and Earth System Sciences
Discussions

the lowest validation errors. Note that an early stopping rule was implemented, that the number of boosting iterations evaluated might be fewer than *nrounds* (see Table 1). The effectiveness of a set of hyperparameters can be measured by its validation errors associated with the optimal number of boosting iterations, which are termed as the inner CV errors.

The prediction errors on different validation folds can vary considerably, as indicated by the width of the ribbons. This suggests that the evaluation results for different sets of hyperparameters could be affected by the randomness of the data splitting methods. Thus, it is important to train and validate the models repeatedly on various splits of the data to understand the uncertainties related to sampling, and to lower the risk of choosing models that simply exploit the random variation of the data. 5-fold CV and the stratified sampling method were used in this study to estimate the model's prediction accuracy, considering the computational efficiency and the stability of the estimate. Data of the same rainfall events were also grouped in the resampling process to avoid data leakage. Other resampling methods, such as repeated K-fold CV, bootstrapping, leave-one-out CV, and leave-group-out CV may also be used, and their ability in identifying high-quality models can be compared (Kuhn and Johnson, 2013). Nevertheless, whether the methods used in this study can identify high-quality models is investigated in Section 3.2. In-depth discussions on resampling methods and model's performance estimate can be found in Bengio and Grandvalet (2004) and Zhang and Yang (2015).

The inner CV errors associated with a set of feature engineering hyperparameters when evaluated together with every set of the XGBoost hyperparameters are shown in Fig. 3b. Although the mean of CV errors (shown as dots) varied for different sets of XGBoost hyperparameters, the differences among them were small when compared to variations of the estimated errors in different inner CV iterations (shown as vertical bars). Thus, the difference in the mean of CV errors may be affected by the randomness of the data, besides the true effect of the hyperparameters. Kuhn and Johnson (2013) suggest the simplest model should be selected among a group of models with similar prediction accuracies, e.g., when the mean of CV errors of a model is within one standard deviation of the errors of the other models. Thus, in this study, in addition to selecting the model with smaller mean inner CV error for each set of feature engineering hyperparameters (i.e., the "best" models), the non-dominated models are also chosen and considered as optimal. Here, the models were ranked according to the error of each inner CV iteration using the fast non-dominated sorting method proposed in Deb et al. (2002). A model is non-dominated if there is no model that performed better in every inner CV iteration among the considered models.

In this study, the values of feature engineering hyperparameters are generated randomly using hydrological knowledge, and those values carry physical meaning. Thus, it is reasonable to assume that if the hyperparameter values and the physical conditions of the site are aligned with each other, those values should result in higher-quality models. This pattern should be consistent for different splits of the data. Figure 4a compares the mean inner CV errors of the models resulted from different sets of feature engineering hyperparameters obtained in different outer CV iterations. The models of different sets of hyperparameters are sorted by their mean inner CV error obtained during the 5 outer CV iterations. An increasing trend can be observed between the inner CV error of each outer iteration and the overall rank of the models, despite the large variations in the estimated inner CV errors among different outer CV iterations. Figure 4b normalizes the differences in the inner CV errors by showing their ranks within each outer CV iteration. It is clear that the effectiveness of different sets of feature

https://doi.org/10.5194/hess-2020-460
Preprint. Discussion started: 7 October 2020


Hydrology and Earth System Sciences
Discussions

engineering hyperparameters is consistent for different splits of data that are used for training and validation, i.e., a high-quality model built on one dataset is likely to perform well on a new dataset. Thus, it is meaningful to optimize the values of the feature engineering hyperparameters, rather than using random values. The consistency also suggests that the resampling method used in this study is appropriate and the data quality is satisfactory, albeit the noticeable variations of different folds related to its small size. The interpretation of the optimal feature engineering hyperparameters is further investigated in Section 3.3.

3.2 Evaluation of fitted models

The optimal XGBoost hyperparameters for each set of feature engineering hyperparameters are found in the previous section through a grid search. Note that the number of models compared is considerably large, as the prediction errors of the models obtained at each boosting iteration are compared in order to identify the optimal number of boosting iterations. Comparing a large number of models on the same validation dataset is subject to overfitting the model selection criterion, which means that the model selected for having good validation performance may be selected by coincidence and may generalize poorly to unseen data (Cawley and Talbot, 2010; Ng, 1997). Cawley and Talbot (2010) suggest that model selection may be considered as an integral part of the model fitting, and it is useful to test the selected model on a holdout dataset. Therefore, this study adopts nested CV procedures to test the quality of the selected models on the test folds. The outer CV error may be considered as an estimate of the model's generalization error.

Section 3.1 reports that the hyperparameter optimization process can be compromised by the considerable variations of the estimated prediction errors at different inner CV iterations, which are related to the randomness involved in resampling small datasets. The outer CV error is also useful in this case for verifying if high-quality models are found through hyperparameter optimization.

The models to be evaluated on the test folds are trained on the training folds (outer) using each set of feature engineering hyperparameters and its optimal XGBoost hyperparameters. Figure 5 compares the mean inner CV error and the mean outer CV error of models built with each set of hyperparameters obtained in the outer CV iterations. These two quantities are positively correlated, suggesting that good models identified in the inner CV iterations generalized well to unseen data. Thus, there is no clear overfitting in the model selection criterion and the hyperparameter optimization procedure is proved to be useful in finding higher quality models. Similar results were obtained for models built with different rainfall depth feature generation options. Further investigation on the topic of overfitting in the model selection criterion in hydrological modeling is recommended.

The estimated inner and outer CV errors varied considerably at different outer CV iterations, as indicated by the vertical and the horizontal lines in Fig. 5. Although the variations do not affect the conclusion on the effectiveness of the model fitting methods (which is characterized by the mean errors), it is important to keep in mind that the estimated generalization error (i.e., the outer CV error) is uncertain. Smaller variations are expected if more data are available, i.e., when the distributions of the data in the training, validation, and test folds are similar to each other. However, data for machine learning are often limited

https://doi.org/10.5194/hess-2020-460
Preprint. Discussion started: 7 October 2020


Hydrology and Earth System Sciences
Discussions

in hydrological studies (Zheng et al., 2018). The association between the volume of data and the uncertainty in generalization error estimation in hydrological modeling deserves further investigation. The nested CV procedures used in this study are recommended for evaluating the model fitting methods and understanding the uncertainty of the estimated generalization error. Two sets of models are selected at each outer CV iteration, i.e., the "best" model built with the hyperparameters that minimize the inner CV error, and the ensemble of non-dominated models (see Sections 2.2.1 and 3.1 for definition). The hydrographs predicted by these models are compared to the observed hydrographs for large runoff events in the test folds in Fig. 6, where the mean, minimal, and maximum predictions of the model ensembles are shown. The performance metrics indicate that both the results of the best models and the mean of the ensemble models fit well to the observations (Krause et al., 2005). Upon visual inspection, the rising and falling limbs of the hydrographs are well predicted, and low flows are well matched. The models underpredict peak flows for some events. This may be caused by lack of similar instances in the training set. For example, the first runoff event in the first outer CV iteration in Fig. 6 was caused by a rainfall event with a relatively small peak rainfall intensity compared to the other runoff events of similar magnitudes. Thus, the models underpredicted this runoff event. Also, note that the data is not error-free, a gap fill process was performed prior to model training (see Section 2.2.1). Training independent models for high flow predictions may relieve the problem of under-prediction. It would, however, lead to even smaller datasets for model training, and it is hard to determine in which case these high flow models should be used. Nevertheless, the quality of the models examined may be considered satisfactory. Simple model prediction post-processing methods, such as the k-nearest neighbor based method (Wani et al., 2017), may be useful for further improving the prediction accuracy.

The best models and non-dominated models are similar to each other, as shown in Fig. 6. The prediction intervals that characterize the differences among the ensemble members are hardly visible. This indicates that the good quality models built on different features learned to make similar predictions. It is interesting to investigate whether the features of these models share similar traits. This question is studied in Section 3.3.1. The ensemble models did not seem to have advantages over the best models, as they did not improve the prediction accuracy or give useful prediction intervals. Thus, the findings do not support the hypothesis proposed in Section 3.1, that selecting the non-dominated models instead of the best model increases the chance of finding better-generalized models. Note that the conclusion is only applied to this case study. In practice, ensembles can be formed by different types of models, rather than different versions of XGBoost models used in this study. However, as previously mentioned, evaluating too many models is subject to overfitting the model selection criterion, and can be computationally expensive.

The predicted and observed flow rates of the best models for the test folds are compared in Fig. 7a. Besides the high correlations between the predicted and the observed values, the variation of the temporal distribution of the data is also noticeable, especially for the high flows. This is because high flows triggered by large storms are rare in each year. No clear patterns can be detected between the prediction error and the magnitude of the runoff event or the year of occurrence, as shown in Fig. 7b. Thus, a bias-correction model that only looks at the predicted values may not be useful in this case.

https://doi.org/10.5194/hess-2020-460
Preprint. Discussion started: 7 October 2020


Hydrology and Earth System Sciences
Discussions

3.3 Factors influential to prediction accuracy and representations of hydrological processes learned by the models

3.3.1 Correlation between feature engineering hyperparameters and models' prediction accuracy

Different sets of feature engineering hyperparameters were used in previous sections to build rainfall-runoff models. This section builds XGBoost models to investigate the correlation between feature engineering hyperparameters and the quality of the resulting rainfall-runoff models. The models built here are termed as explanation models, the function of which is to infer the physical factors that are related to the runoff generation processes. An explanation model was trained for each outer CV iteration that was used for training the rainfall-runoff models in previous sections. The input variable is the feature engineering hyperparameters (Table 1), and the output is the estimated inner root-mean-square error (RMSE) of the "best" model. 100 samples were available for each explanation model (corresponding to 100 sets of feature engineering hyperparameters). Each model was tuned using a 10-fold CV, repeated 5 times, where 90 samples were used for training and validation, and 10 samples were reserved for testing. The models generally had good fits to the data. The goodness-of-fit of the model of outer CV iteration #3 is shown in Fig. 8a as an example.

The SHAP values were computed for each sample and each model. The results for the model of outer CV iteration #3 are shown in Fig. 8b. l has the largest impact on prediction accuracy, as shown by the mean SHAP value on the y-axis. The other features had smaller impacts on the prediction accuracy, but their ranks were different for models of different outer CV iterations (not shown). Some interesting patterns can be detected in the correlation between SHAP value and are consistent for all the models at different outer CV iterations. For example, smaller l is associated with lower SHAP value, corresponding to decreases in inner CV RMSE. Positive SHAP values are associated with $account\_CumRain = 1$, suggesting that prediction accuracy decreased when accumulated rainfall depth since the beginning of observation was used as an input variable. The $account\_seaon$ feature has little impact on the model output. The dependence between the feature values and SHAP values are also depicted in Fig. 8c. The color of the dots represents the predicted inner CV RMSE which has no obvious correlations with the feature values. This implies that the predicted inner CV RMSE is not dominated by a single feature. Thus, building models involving all the features is essential for understanding their impact on the inner CV RMSE. The other patterns shown in Fig. 8c, except the positive correlation between l and the inner CV RMSE, are inconsistent across models built for different outer CV iterations (not shown).

The SHAP values and the dependence information can be used to understand the models' behavior and optimize hyperparameters. For instance, new models can be built for WS with smaller l, $account\_CumRain = 0$, and $account\_season = 0$. These models are expected to deliver good prediction accuracies. However, this needs to be confirmed using test datasets (Kuhn and Johnson, 2013). The results also indicate that the runoff is most related to recent rainfalls (i.e., smaller l), and the model quality deteriorates when redundant variables are used in modeling. The impact of $account\_CumRain$ and $account\_season$ on the runoff generation process is not profound. The conclusion is, of course, affected by our ability to estimate the true prediction accuracy and the accuracy of the explanation models. Also, note that the redundant variables in XGBoost may be useful in other machine learning methods. Further confirmation of the inferred

https://doi.org/10.5194/hess-2020-460
Preprint. Discussion started: 7 October 2020


Hydrology and Earth System Sciences
Discussions

correlations between features and runoff response requires an evaluation of different types of methods and reasoning with hydrological theories.

3.3.2 Hydrolgoical processes according to the machine learning models

The SHAP values were computed for each rainfall depth feature $D_{t-a,t-b}$ and each prediction. The average SHAP value across all predictions was computed for each $D_{t-a,t-b}$, which was then distributed equally among the rainfall depth variables P_{t-i} recorded between time $t-a$ and $t-b$. Such value computed for P_{t-i} specifies the average contribution of P_{t-i} to the runoff at time step t, which may also be interpreted as the average contribution of P_t to the runoff i time steps into the future. The average contributions of P_t for runoffs at different time steps ahead in various models are shown in Fig. 9. Each time step is 10 minutes. Each subfigure corresponds to the results of the models obtained during a specific outer CV iteration. All the models find P_t to have higher impacts on the runoffs of the next few time steps. The impact decreased rapidly for runoffs that are further into the future (note that x-axis is in pseudo-logarithm scale), suggesting that the catchment does not exhibit a strong "memory effect" (Kratzert et al., 2018) and the temporal scales involved in the runoff generation processes are short. The best models and the mean of ensemble models (as defined in Section 3.2) suggest that P_t has the highest impact on the runoffs at one time step ahead. This result indicates that the average response time of the catchment is about one time step, i.e., 10 minutes, which is reasonable considering its small size.

The average contributions assigned by the ensemble models varied noticeably in different outer CV iterations, which was caused by random sampling of feature engineering hyperparameters. Nevertheless, the variations in the SHAP values are canceled out when the mean of the ensemble models is used for prediction (note that SHAP values are additive). The contribution values assigned by the best model are similar to that of the mean of the ensemble models, suggesting that the best model found by minimizing the mean inner CV error can be regarded as an average of the plausible explanations of rainfall-runoff processes of the catchment. The outer CV errors of the ensemble models are shown as the intensity of the color in Fig. 9, where no clear correlations between the assigned SHAP values and the estimated generalization error can be detected, as indicated by the small variations in the intensity of color in each subfigure.

The SHAP values allow localized explanation for each prediction. As SHAP values satisfy local accuracy property, it is possible to decompose a flow rate prediction into flow rates contributed by each feature. This is useful for hydrograph separation (Pelletier and Andréassian, 2020), examples of which for large, medium, and small runoff events are shown in Fig. 10. The model analyzed here is the best model of outer CV iteration #3.

The peak runoffs in large storm events are mostly controlled by recent rainfalls, which may be caused by the overflows from the rain garden and the fast stormwater infiltration through the porous pavement. The recession leaps of the hydrograph are affected by rainfalls of the relatively distant past. This may be explained by the slow infiltration process of the rain garden and soils. For the medium and smaller events, the peak runoffs are also related to recent rainfalls, but the contributions of the past rainfalls are substantial. This pattern may correspond to cases that runoffs are triggered by small rainfall events when the

https://doi.org/10.5194/hess-2020-460
Preprint. Discussion started: 7 October 2020


Hydrology and Earth System Sciences
Discussions

storage capacity of the catchment is nearly saturated, such that the rainfalls in the past also play important roles in regulating the flow rate. It is also interesting to note that some rainfall depth records actually have negative contributions to runoff, suggesting a deficit in catchment wetness. This deficit is pronounced at the beginning of runoff events, which may be caused by extended dry periods. The deficit becomes smaller as the rainfall event proceeds. For the small runoff events, the deficits are also related to recent rainfalls, which may be an indication of insufficient runoffs generated through the fast runoff routes (i.e., overflows and fast infiltration through porous pavements). The bias shown in the four subfigures is a constant value throughout all time steps, which is the prediction made by the model when there is no information on any features. The models examined here only use rainfall depth features as input, i.e.,$account\_CumRain = 0$, and $account\_season = 0$. In cases that other features are used as input, their contributions may be considered as an enlargement or reduction factor to the contributions of the rainfalls. Lundberg et al. (2020) also provides methods to explicitly analyze the interactions between two features.

The contribution P_t has on future runoffs can be computed using Eq. (11). The continuing impact of the peak rainfall records on runoffs is shown in Fig. 11, where the four runoff events are the same as in Fig. 10. The contributions of all other factors are aggregated to highlight the contributions of the peak rainfall. The peak rainfall's contribution decreases rapidly for runoffs that are further into the future. The impact of peak rainfalls lasts for about 30 to 50 time steps or 300 to 500 minutes, which is consistent with the patterns shown in Fig. 9. The results obtained here are useful for understanding the drainage processes, e.g., analyzing the impact of the rainfalls in certain periods on future runoffs, and computing the runoff coefficient associated with each rain depth record. As a method to decompose flow rate to contributions of rainfalls at each time step is provided in Eq. (11), it is possible to determine the "water age" of the stormwater runoff at each time step.

It is important to note here that the explanations of the involved hydrological processes are essentially modeling results and may not match the physical processes. The models are affected by prediction errors, and it is currently not clear whether the truthfulness of the explanations of a model is associated with its generalization error. This study roughly compares the hydrograph composition of the best models obtained at different outer CV iterations (not shown) and find similar patterns among them. It would be meaningful to compare the hydrological processes learned by the machine learning models to the physical measurement or the processes inferred by the process-based models (which is also subject to model uncertainty involved in parameter estimation). Such tasks are not performed in this study because of the limited amount of data available for setting up a process-based model, which is also a motivation for using machine learning methods. Nevertheless, this study shows that XGBoost models built using nested CV and a feature engineering scheme provide plausible explanations to the involved hydrological processes, and each runoff prediction can be explained, offering modelers the opportunity to investigate the hydrological properties of the catchment and justify each runoff prediction using hydrological knowledge.

3.4 Comparison of different models

This section compares the XGBoost models trained with different rainfall depth feature generation options for WS (previous sections only present results of option 1 models in detail). Linear regression models, as the baseline models, are also examined here. The candidate feature engineering hyperparameters (Table 1) and the CV folds are the same for all the models. The inner

https://doi.org/10.5194/hess-2020-460
Preprint. Discussion started: 7 October 2020


Hydrology and Earth System Sciences
Discussions

and outer CV errors of the models are shown in Table 2. Models built with different options are found to have similar prediction accuracies. The standard deviations of the outer CV RMSE are larger than that of the inner CV fold. This may be a result of fewer samples in the test fold for estimating the outer CV error, such that the evaluation results are more affected by the randomness of sampling. The results of the linear regression models are almost identical, with differences in RMSE smaller than 0.001. This is because the features resulted from a feature creation option can be represented as a linear combination of that from another option. Thus, each P_t in linear regression models trained with the same hyperparameters but different rainfall depth feature creation options were essentially assigned with the same weight. The linear regression models generally performed worse than the XGBoost models, suggesting that it is meaningful to adopt more complex machine learning methods, such as XGBoost, to acquire models with higher prediction accuracies, given that the interpretability of the complex methods is greatly enhanced by model explanation methods, such as SHAP and LIME (Molnar, 2020).

XGBoost models and linear regression models are built for SHC. The feature engineering and XGBoost hyperparameters are the same as WS, except that $account\_CumRain$ and $account\_season$ are always set to be zero (see explanations in Section 2.2.2). The hydrographs predicted by the XGBoost models and the SWMM model built in Lee et al. (2018a) for the test period are shown in Fig. 12. The hydrographs predicted by XGBoost generally match the observed hydrograph. The performance metrics of different models are shown in Fig. 12 and Table 3. Linear regression models are found to have worse performance compared to the other models. The performance of XGBoost models is satisfactory, considering that only one month of data were used for model training. This case study shows that the proposed model training methods can be applied in catchments of various conditions, and little adjustment is required.

4 Conclusions

This study uses machine learning methods to build high-temporal resolution rainfall-runoff models for two urban catchments with different sizes, drainage system configurations, and data availabilities. A simple and physically meaningful feature engineering method that extracts useful features from high-resolution rainfall time series is proposed. This method, when used in combination with XGBoost methods, hyperparameter optimization methods, and nested cross-validation procedures, are found to be able to generate accurate models for runoff prediction. Only rainfall-runoff data are used in the model training processes in this study, while other potentially important hydro-environmental factors may also be incorporated into modeling. The proposed model training methods are semi-automatic, requiring minimal user input. The problems of having insufficient information for setting up process-based models are thus addressed. The SHAP method is used to interpret the predictions made by the models, for which the predicted runoff at each time step is decomposed into the contributions of the rainfalls at different time steps. The results are further utilized for catchment response time estimation, hydrograph separation, and analysis of the continuing effect of rainfall events. The SHAP method is also used in this study to identify the factors influential to the rainfall-runoff relationship.

The contribution of this study is three-fold.

https://doi.org/10.5194/hess-2020-460
Preprint. Discussion started: 7 October 2020


Hydrology and Earth System Sciences
Discussions

(1) This study demonstrates that modern machine learning methods, when properly implemented, are useful for predicting the hydrological responses of SuDS in urban catchments of different conditions at fine temporal scales.

(2) Model explanation methods are used in this study to investigate the basis on which machine learning models make predictions. Furthermore, a method to assign the contributions of rainfall at each time step to runoffs at different steps is proposed and is used in various applications, such as hydrograph separation and estimation of catchment response time. This study shows that machine learning methods can discover plausible representations of physical processes, allowing them to be useful for studying the hydrological processes that govern the interested input-output relationships.

(3) This study presents a complete machine learning framework for urban hydrological studies, which includes feature engineering methods, hyperparameter optimization methods, generalization error estimation methods, and model explanation methods. The framework is particularly useful for urban catchments where the information for setting up process-based models is insufficient.

This study presents an initial investigation into the topic of modeling and interpreting the urban hydrological processes at fine temporal scales using machine learning methods. A few findings emerged from the case studies. First, the feature engineering methods and nested cross-validation procedures are found to be useful for building higher quality models and estimating generalization errors. The feature engineering methods, however, can be designed arbitrarily, and evaluating the effectiveness of many feature engineering methods can be computationally expensive and subject to overfitting the model selection criterion. Thus, future studies can explore the application of machine learning methods that rely less on feature engineering, such as deep learning. Second, this study shows that randomness in resampling results in considerable uncertainty in the generalization estimation. Future studies are recommended to adopt the nested cross-validation procedures to explicitly investigate the uncertainties and to avoid overfitting. Third, the representation of the hydrological processes learned by models are found to be plausible. It is recommended to use the SHAP method in more case studies and compare the representations learned by different models. The representation of hydrological processes learned by machine learning models may also be compared with physical measurement or the representation of process-based models. The correlation between the truthfulness of model explanation and the generalization errors can be investigated in future research. Finally, using model explanation methods, such as SHAP, allows the reasons for making each prediction to be explained and the learned representations to be analyzed, which increases the transparency of machine learning, and may eventually promote the application of machine learning methods in hydrological studies.

Code availability

The source code used for the machine learning model training and interpretation methods presented in the paper is available at https://github.com/stsfk/explainable_ml_hydro, where an example project is provided. The XGBoost and SHAPforxgboost packages in R are used in this research, both of which are freely available.

https://doi.org/10.5194/hess-2020-460
Preprint. Discussion started: 7 October 2020

Data availability

The data of the two case studies used in this research is obtained from the United States Geological Survey (USGS), Clermont County, Ohio, the U.S., and the United States Environmental Protection Agency (US EPA). The SWMM model used in this research is developed in Lee et al. (2018a).

Author contribution

YY designed the study, acquired the data, wrote the code, analyzed the results, and prepared the manuscript. TFMC contributed to the design of numerical experiments, supervised the study, validated the results, and revised the manuscript.

Competing interests

The authors declare that they have no conflict of interest.

Acknowledgments

The work described in this paper was partly supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. HKU17255516), and partly supported by the RGC Theme-based Research Scheme (Grant No: T21-711/16-R) funded by the Research Grants Council of the Hong Kong Special Administrative Region, China. We thank Robert Darner from USGS, Christ Nietch from USEPA, Joong Gwang Lee from Center for Urban Green Infrastructure Engineering, and Bill Mellman from Clermont County Water Resources for providing data for this research.

References

Bengio, Y. and Grandvalet, Y.: No unbiased estimator of the variance of K-fold cross-validation, J. Mach. Learn. Res., 5, 1089–1105, 2004.

Beven, K.: Rainfall-Runoff Modelling, John Wiley & Sons, Ltd, Chichester, UK., 2012.

Biecek, P. and Burzykowski, T.: Explanatory Model Analysis Explore, Explain and Examine Predictive Models. [online] Available from: https://pbiecek.github.io/ema/ (Accessed 30 June 2020), 2020.

Cawley, G. C. and Talbot, N. L. C.: On over-fitting in model selection and subsequent selection bias in performance evaluation, J. Mach. Learn. Res., 11, 2079–2107 [online] Available from: http://www.jmlr.org/papers/volume11/cawley10a/cawley10a.pdf (Accessed 29 August 2019), 2010.

Charlesworth, S. M.: A review of the adaptation and mitigation of global climate change using sustainable drainage in cities, J. Water Clim. Chang., 1(3), 165–180, doi:10.2166/wcc.2010.035, 2010.

https://doi.org/10.5194/hess-2020-460
Preprint. Discussion started: 7 October 2020


Hydrology and Earth System Sciences
Discussions

Chen, T. and Guestrin, C.: XGBoost: A scalable tree boosting system, in Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, vol. 13-17-Augu, pp. 785–794., 2016.

Chen, T. and He, T.: xgboost: eXtreme Gradient Boosting, [online] Available from: https://cran.r-project.org/web/packages/xgboost/vignettes/xgboost.pdf (Accessed 29 June 2020), 2020.

Chollet, F. and Allaire, J. J.: Deep Learning with R., 2018.

Damodaram, C., Giacomoni, M. H., Prakash Khedun, C., Holmes, H., Ryan, A., Saour, W. and Zechman, E. M.: Simulation of combined best management practices and low impact development for sustainable stormwater management, J. Am. Water Resour. Assoc., 46(5), 907–918, doi:10.1111/j.1752-1688.2010.00462.x, 2010.

Darner, R. A. and Dumouchelle, D. H.: Hydraulic Characteristics of Low-Impact Development Practices in Northeastern Ohio, 2008-2010: U.S. Geological Survey Scientific Investigations Report 2011–5165. [online] Available from: https://pubs.usgs.gov/sir/2011/5165/ (Accessed 7 July 2020), 2011.

Darner, R. A., Shuster, W. D. and Dumouchelle, D. H.: Hydrologic Characteristics of Low-Impact Stormwater Control Measures at Two Sites in Northeastern Ohio , 2008 – 13: U.S. Geological Survey Scientific Investigations Report 2015-5030., 2015.

Deb, K., Pratap, A., Agarwal, S. and Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Trans. Evol. Comput., 6(2), 182–197, doi:10.1109/4235.996017, 2002.

DeBusk, K. M., Hunt, W. F. and Line, D. E.: Bioretention Outflow: Does It Mimic Nonurban Watershed Shallow Interflow?, J. Hydrol. Eng., 16(3), 274–279, doi:10.1061/(ASCE)HE.1943-5584.0000315, 2011.

Eckart, K., McPhee, Z. and Bolisetti, T.: Performance and implementation of low impact development – A review, Sci. Total Environ., 607–608, 413–432, doi:10.1016/j.scitotenv.2017.06.254, 2017.

Eggimann, S., Mutzner, L., Wani, O., Schneider, M. Y., Spuhler, D., Moy De Vitry, M., Beutler, P. and Maurer, M.: The Potential of Knowing More: A Review of Data-Driven Urban Water Management, Environ. Sci. Technol., 51(5), 2538–2553, doi:10.1021/acs.est.6b04267, 2017.

Elliott, A. H. and Trowsdale, S. A.: A review of models for low impact urban stormwater drainage, Environ. Model. Softw., 22(3), 394–405, doi:10.1016/j.envsoft.2005.12.005, 2007.

Eric, M., Li, J. and Joksimovic, D.: Performance Evaluation of Low Impact Development Practices Using Linear Regression, Br. J. Environ. Clim. Chang., 5(2), 78–90, doi:10.9734/bjecc/2015/11578, 2015.

Fassman-Beck, E., Hunt, W., Berghage, R., Carpenter, D., Kurtz, T., Stovin, V. and Wadzuk, B.: Curve number and runoff coefficients for extensive living roofs, J. Hydrol. Eng., 21(3), 04015073, doi:10.1061/(ASCE)HE.1943-5584.0001318, 2016.

Fletcher, T. D., Shuster, W., Hunt, W. F., Ashley, R., Butler, D., Arthur, S., Trowsdale, S., Barraud, S., Semadeni-Davies, A., Bertrand-Krajewski, J. L., Mikkelsen, P. S., Rivard, G., Uhl, M., Dagenais, D. and Viklander, M.: SUDS, LID, BMPs, WSUD and more – The evolution and application of terminology surrounding urban drainage, Urban Water J., 12(7), 525–542, doi:10.1080/1573062X.2014.916314, 2015.

Friedman, J. H.: Greedy function approximation: A gradient boosting machine, Ann. Stat., 29(5), 1189–1232,

doi:10.1214/aos/1013203451, 2001.

Gimenez-Maranges, M., Breuste, J. and Hof, A.: Sustainable Drainage Systems for transitioning to sustainable urban flood management in the European Union: A review, J. Clean. Prod., 255, 120191, doi:10.1016/j.jclepro.2020.120191, 2020.

Guo, Y. and Senior, M. J.: Climate model simulation of point rainfall frequency characteristics, J. Hydrol. Eng., 11(6), 547–554, doi:10.1061/(ASCE)1084-0699(2006)11:6(547), 2006.

Jang, W. S., Engel, B. and Yeum, C. M.: Integrated environmental modeling for efficient aquifer vulnerability assessment using machine learning, Environ. Model. Softw., 124, 104602, doi:10.1016/j.envsoft.2019.104602, 2020.

Johannessen, B. G., Hanslin, H. M. and Muthanna, T. M.: Green roof performance potential in cold and wet regions, Ecol. Eng., 106, 436–447, doi:10.1016/j.ecoleng.2017.06.011, 2017.

Jones, P. and Macdonald, N.: Making space for unruly water: Sustainable drainage systems and the disciplining of surface runoff, Geoforum, 38(3), 534–544, doi:10.1016/j.geoforum.2006.10.005, 2007.

Karbasi, M.: Forecasting of Multi-Step Ahead Reference Evapotranspiration Using Wavelet- Gaussian Process Regression Model, Water Resour. Manag., 32(3), 1035–1052, doi:10.1007/s11269-017-1853-9, 2018.

Khan, U. T., Valeo, C., Chu, A. and He, J.: A data driven approach to bioretention cell performance: Prediction and design, Water (Switzerland), 5(1), 13–28, doi:10.3390/w5010013, 2013.

Kratzert, F., Klotz, D., Brenner, C., Schulz, K. and Herrnegger, M.: Rainfall-runoff modelling using Long Short-Term Memory (LSTM) networks, Hydrol. Earth Syst. Sci., 22(11), 6005–6022, doi:10.5194/hess-22-6005-2018, 2018.

Kratzert, F., Herrnegger, M., Klotz, D., Hochreiter, S. and Klambauer, G.: NeuralHydrology – Interpreting LSTMs in Hydrology, in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 11700 LNCS, pp. 347–362., 2019.

Krause, P., Boyle, D. P. and Bäse, F.: Comparison of different efficiency criteria for hydrological model assessment, Adv. Geosci., 5, 89–97, doi:10.5194/adgeo-5-89-2005, 2005.

Kuhn, M. and Johnson, K.: Applied predictive modeling., 2013.

Kuhn, M. and Johnson, K.: Feature Engineering and Selection : a Practical Approach for Predictive Models., Chapman and Hall/CRC. [online] Available from: https://www.routledge.com/Feature-Engineering-and-Selection-A-Practical-Approach-for-Predictive-Models/Kuhn-Johnson/p/book/9781138079229 (Accessed 24 July 2020), 2019.

Lee, J. G., Nietch, C. T. and Panguluri, S.: Drainage area characterization for evaluating green infrastructure using the Storm Water Management Model, Hydrol. Earth Syst. Sci., 22(5), 2615–2635, doi:10.5194/hess-22-2615-2018, 2018a.

Lee, J. G., Nietch, C. T. and Panguluri, S.: SWMM Modeling Methods for Simulating Green Infrastructure at a Suburban Headwatershed: User's Guide, U.S. Environ. Prot. Agency, (October), 157 [online] Available from: https://nepis.epa.gov/Exe/ZyPDF.cgi/P100TJ39.PDF?Dockey=P100TJ39.PDF%0A (Accessed 11 July 2020b), 2018.

Levin, L. A. and Mehring, A. S.: Optimization of bioretention systems through application of ecological theory, Wiley Interdiscip. Rev. Water, 2(3), 259–270, doi:10.1002/wat2.1072, 2015.

Li, S., Kazemi, H. and Rockaway, T. D.: Performance assessment of stormwater GI practices using artificial neural networks,

https://doi.org/10.5194/hess-2020-460
Preprint. Discussion started: 7 October 2020

Sci. Total Environ., 651, 2811–2819, doi:10.1016/j.scitotenv.2018.10.155, 2019.

Liu, J., Sample, D., Bell, C. and Guan, Y.: Review and Research Needs of Bioretention Used for the Treatment of Urban Stormwater, Water, 6(4), 1069–1099, doi:10.3390/w6041069, 2014.

Lundberg, S. M. and Lee, S. I.: A unified approach to interpreting model predictions, in Advances in Neural Information Processing Systems, vol. 2017-Decem, pp. 4766–4775. [online] Available from: https://github.com/slundberg/shap (Accessed 30 June 2020), 2017.

Lundberg, S. M., Erion, G. G. and Lee, S.-I.: Consistent Individualized Feature Attribution for Tree Ensembles, [online] Available from: http://github.com/slundberg/shap (Accessed 6 July 2020), 2018.

Lundberg, S. M., Erion, G., Chen, H., DeGrave, A., Prutkin, J. M., Nair, B., Katz, R., Himmelfarb, J., Bansal, N. and Lee, S.-I.: From local explanations to global understanding with explainable AI for trees, Nat. Mach. Intell., 2(1), 56–67, doi:10.1038/s42256-019-0138-9, 2020.

Maier, H. R. and Dandy, G. C.: Neural networks for the prediction and forecasting of water resources variables: a review of modelling issues and applications, Environ. Model. Softw., 15(1), 101–124, doi:10.1016/S1364-8152(99)00007-9, 2000.

Massoudieh, A., Maghrebi, M., Kamrani, B., Nietch, C., Tryby, M., Aflaki, S. and Panguluri, S.: A flexible modeling framework for hydraulic and water quality performance assessment of stormwater green infrastructure, Environ. Model. Softw., 92, 57–73, doi:10.1016/j.envsoft.2017.02.013, 2017.

Mitchell, R. and Frank, E.: Accelerating the XGBoost algorithm using GPU computing, PeerJ Comput. Sci., 2017(7), e127, doi:10.7717/peerj-cs.127, 2017.

Molnar, C.: Interpretable Machine Learning, Leanpub. [online] Available from: https://leanpub.com/interpretable-machine-learning (Accessed 30 June 2020), 2020.

Montalto, F., Behr, C., Alfredo, K., Wolf, M., Arye, M. and Walsh, M.: Rapid assessment of the cost-effectiveness of low impact development for CSO control, Landsc. Urban Plan., 82(3), 117–131, doi:10.1016/j.landurbplan.2007.02.004, 2007.

Ng, A. Y.: Preventing Overfitting of Cross-Validation Data, in Proceedings of the Fourteenth International Conference on Machine Learning. [online] Available from: http://ai.stanford.edu/~ang/papers/cv-final.pdf (Accessed 7 September 2019), 1997.

Niazi, M., Nietch, C., Maghrebi, M., Jackson, N., Bennett, B. R., Tryby, M. and Massoudieh, A.: Storm Water Management Model: Performance Review and Gap Analysis, J. Sustain. Water Built Environ., 3(2), 04017002, doi:10.1061/JSWBAY.0000817, 2017.

Nielsen, A.: Practical Time Series Analysis Preview Edition, O'Reilly Media, Inc. [online] Available from: https://www.oreilly.com/library/view/practical-time-series/9781492041641/ (Accessed 30 June 2020), 2019.

Nielsen, D.: Tree Boosting With XGBoost: Why does XGBoost win every machine learning competition?, Master's Thesis, Norweigian Univ. Sci. Technol., (December), 2016, doi:10.1111/j.1758-5899.2011.00096.x, 2016.

Pappalardo, V. and La Rosa, D.: Policies for sustainable drainage systems in urban contexts within performance-based planning approaches, Sustain. Cities Soc., 52, 101830, doi:10.1016/j.scs.2019.101830, 2020.

https://doi.org/10.5194/hess-2020-460
Preprint. Discussion started: 7 October 2020

730 Pelletier, A. and Andréassian, V.: Hydrograph separation: An impartial parametrisation for an imperfect method, Hydrol. Earth Syst. Sci., 24(3), 1171–1187, doi:10.5194/hess-24-1171-2020, 2020.

Platz, M., Simon, M. and Tryby, M.: Testing of the Storm Water Management Model Low Impact Development Modules, J. Am. Water Resour. Assoc., 56(2), 283–296, doi:10.1111/1752-1688.12832, 2020.

Refsgaard, J. C. and Storm, B.: Construction, Calibration And Validation of Hydrological Models, pp. 41–54, Springer,
735 Dordrecht., 1990.

Rosa, D. J., Clausen, J. C. and Dietz, M. E.: Calibration and Verification of SWMM for Low Impact Development, J. Am. Water Resour. Assoc., 1–12, doi:10.1111/jawr, 2015.

Rossman, L. A.: Storm Water Management Model User's Manual Version 5.1 . [online] Available from: https://www.epa.gov/water-research/storm-water-management-model-swmm-version-51-users-manual (Accessed 7 July
740 2020), 2015.

Rossman, L. A. and Huber, W. C.: Storm Water Management Model Reference Manual Volume III – Water Quality., 2016.

Schuol, J. and Abbaspour, K. C.: Calibration and uncertainty issues of a hydrological model (SWAT) applied to West Africa, Adv. Geosci., 9, 137–143, doi:10.5194/adgeo-9-137-2006, 2006.

Sean, L. (George M. U.: Essentials ofMetaheuristics, second edition, Lulu. [online] Available from:
745 http://cs.gmu.edu/↩sean/book/metaheuristics/%0ASean (Accessed 11 July 2020), 2013.

Selbig, W. R., Buer, N. and Danz, M. E.: Stormwater-quality performance of lined permeable pavement systems, J. Environ. Manage., 251, doi:10.1016/j.jenvman.2019.109510, 2019.

Serrano, R.: Cooperative Games : Core and Shapley Value, CEMFI Work. Pap., 2007(0709), 1–22, 2007.

Solomatine, D. P. and Dulal, K. N.: Model trees as an alternative to neural networks in rainfall-runoff modelling, Hydrol. Sci.
750 J., 48(3), 399–411, doi:10.1623/hysj.48.3.399.45291, 2003.

Solomatine, D. P. and Ostfeld, A.: Data-driven modelling: Some past experiences and new approaches, in Journal of Hydroinformatics, vol. 10, pp. 3–22, IWA Publishing., 2008.

Solomatine, D. P., Maskey, M. and Shrestha, D. L.: Instance-based learning compared to other data-driven methods in hydrological forecasting, Hydrol. Process., 22(2), 275–287, doi:10.1002/hyp.6592, 2008.

755 Trinh, D. H. and Chui, T. F. M.: Assessing the hydrologic restoration of an urbanized area via an integrated distributed hydrological model, Hydrol. Earth Syst. Sci., 17(12), 4789–4801, doi:10.5194/hess-17-4789-2013, 2013.

Verleysen, M. and François, D.: The curse of dimensionality in data mining and time series prediction, in Lecture Notes in Computer Science, vol. 3512, pp. 758–770, Springer Verlag., 2005.

Wani, O., Beckers, J. V. L., Weerts, A. H. and Solomatine, D. P.: Residual uncertainty estimation using instance-based learning
760 with applications to hydrologic forecasting, Hydrol. Earth Syst. Sci., 21(8), 4021–4036, doi:10.5194/hess-21-4021-2017, 2017.

Worland, S. C., Farmer, W. H. and Kiang, J. E.: Improving predictions of hydrological low-flow indices in ungaged basins using machine learning, Environ. Model. Softw., 101, 169–182, doi:10.1016/j.envsoft.2017.12.021, 2018.

Xia, Y., Liu, C., Li, Y. Y. and Liu, N.: A boosted decision tree approach using Bayesian hyper-parameter optimization for

credit scoring, Expert Syst. Appl., 78, 225–241, doi:10.1016/j.eswa.2017.02.017, 2017.

Yang, Y. and Chui, T. F. M.: Integrated hydro-environmental impact assessment and alternative selection of low impact development practices in small urban catchments, J. Environ. Manage., 223, 324–337, doi:10.1016/j.jenvman.2018.06.021, 2018.

Yang, Y. and Chui, T. F. M.: Hydrologic Performance Simulation of Green Infrastructures: Why Data-Driven Modelling Can Be Useful?, in New Trends in Urban Drainage Modelling, pp. 480–484, Springer International Publishing., 2019.

Yong, C. F., McCarthy, D. T. and Deletic, A.: Predicting physical clogging of porous and permeable pavements, J. Hydrol., 481, 48–55, doi:10.1016/j.jhydrol.2012.12.009, 2013.

Zeng, X. and Martinez, T. R.: Distribution-balanced stratified cross-validation for accuracy estimation, J. Exp. Theor. Artif. Intell., 12(1), 1–12, doi:10.1080/095281300146272, 2000.

Zhang, K. and Chui, T. F. M.: A review on implementing infiltration-based green infrastructure in shallow groundwater environments: Challenges, approaches, and progress, J. Hydrol., 579, 124089, doi:10.1016/j.jhydrol.2019.124089, 2019.

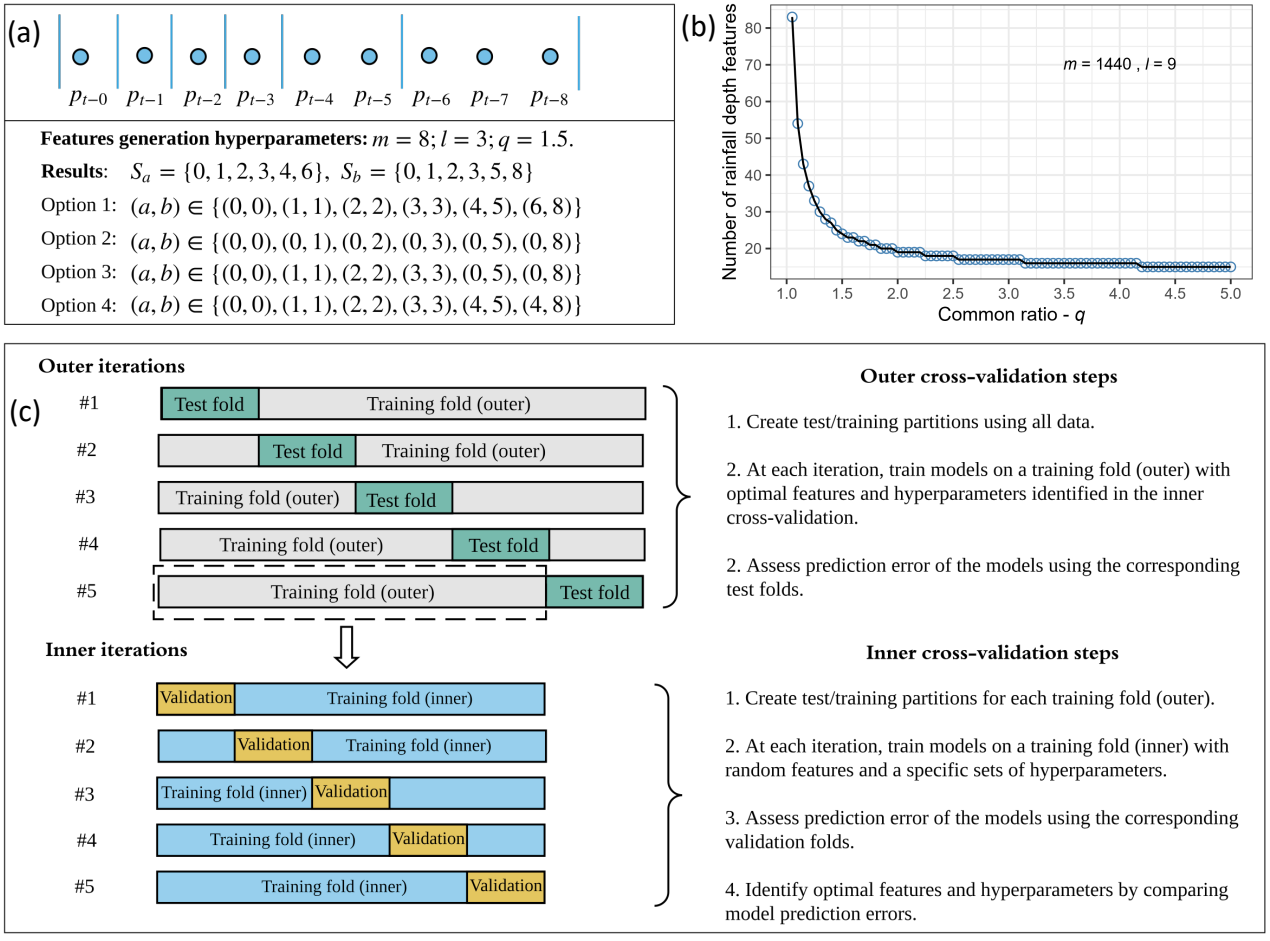
Zhang, Y. and Yang, Y.: Cross-validation for selecting a model selection procedure, J. Econom., 187(1), 95–112, doi:10.1016/j.jeconom.2015.02.006, 2015.

Zheng, F., Maier, H. R., Wu, W., Dandy, G. C., Gupta, H. V. and Zhang, T.: On Lack of Robustness in Hydrological Model Development Due to Absence of Guidelines for Selecting Calibration and Evaluation Data: Demonstration for Data-Driven Models, Water Resour. Res., 54(2), 1013–1030, doi:10.1002/2017WR021470, 2018.

Zhou, Q.: A Review of Sustainable Urban Drainage Systems Considering the Climate Change and Urbanization Impacts, Water, 6(4), 976–992, doi:10.3390/w6040976, 2014.

https://doi.org/10.5194/hess-2020-460
Preprint. Discussion started: 7 October 2020

Hydrology and Earth System Sciences
Discussions
Open Access
EGU

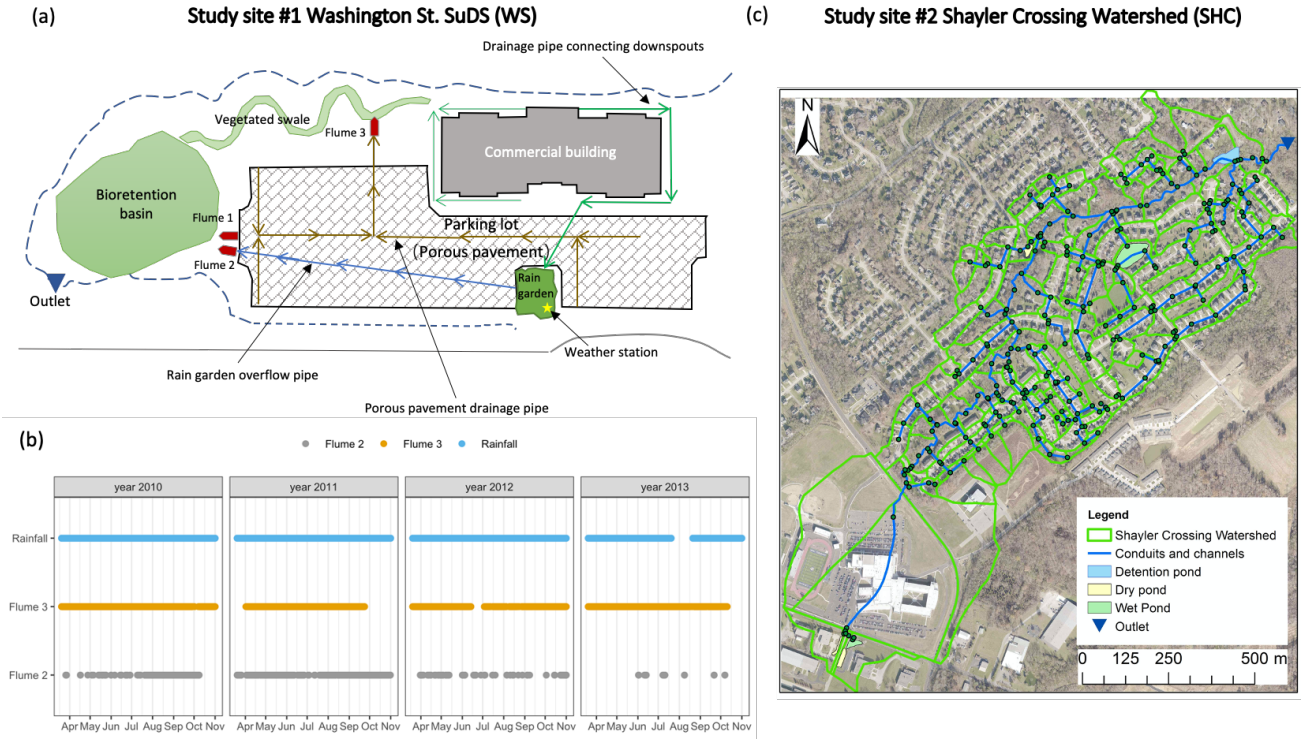


785

Figure 1 (a) Example of the indexes for creating rainfall depth features obtained using a set of feature engineering hyperparameters and different index selection options. (b) The number of rainfall depth features generated when m=1440, l=9, and q varies. (c) The nested cross-validation procedures.

https://doi.org/10.5194/hess-2020-460
Preprint. Discussion started: 7 October 2020

Hydrology and Earth System Sciences
Discussions
Open Access
EGU



790

Figure 2 (a) Layout of the SuDS and monitoring network in Washington Street site (WS), Geauga County, Ohio, the U.S. This figure is adapted from Darner et al. (2015). (b) Availability of the rainfall and flow rate records taken at the regular 10-minute intervals. (c) Map of the Shayler Crossing Watershed (SHC). Subcatchment boundaries and the drainage system shown on the map are defined by Lee et al. (2018a).

795

https://doi.org/10.5194/hess-2020-460
Preprint. Discussion started: 7 October 2020

Hydrology and Earth System Sciences
Discussions

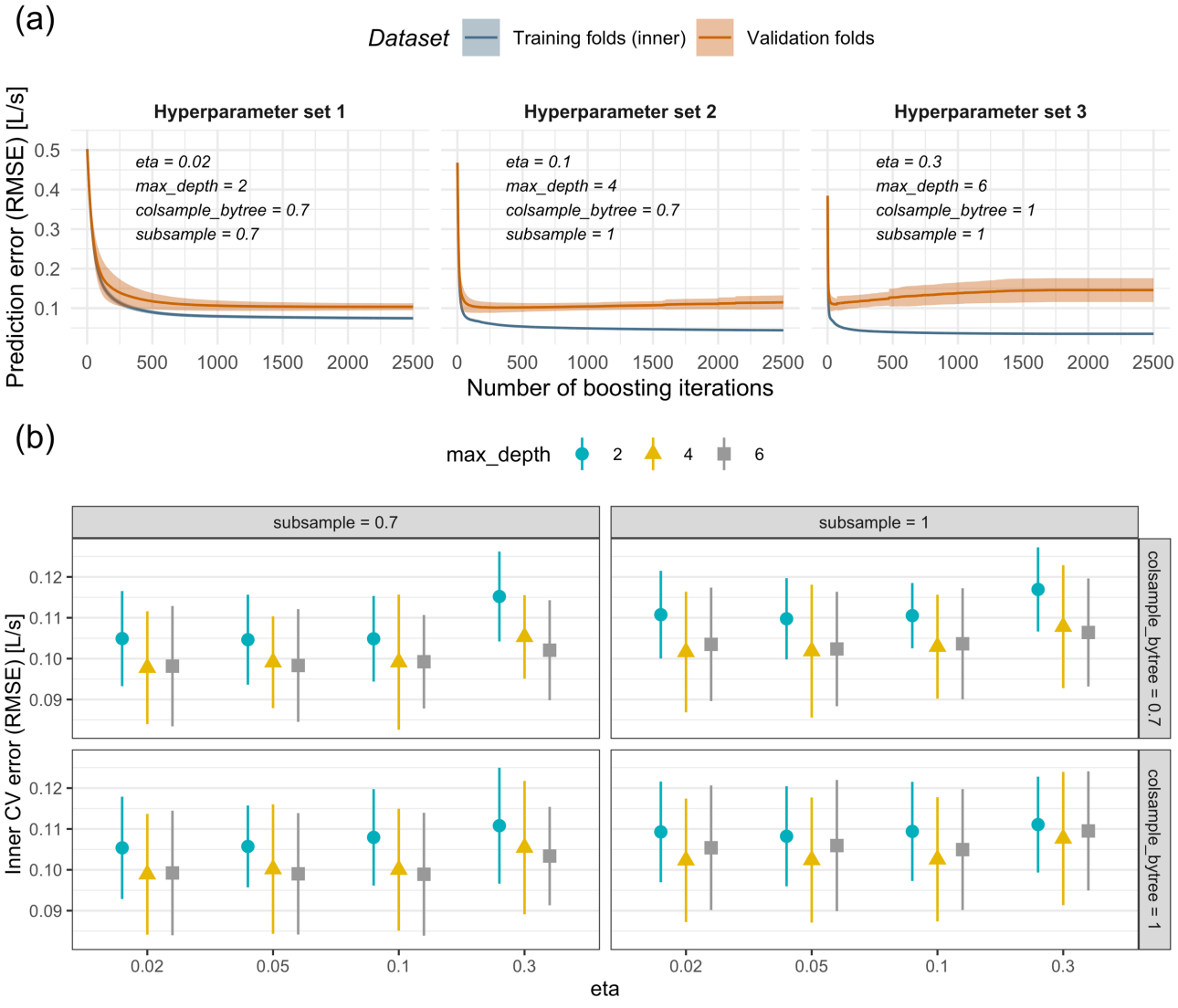


Figure 3 (a) Model's prediction errors on the training and the validation folds when trained using different numbers of boosting iterations. Solid lines correspond to mean values obtained at the inner CV iterations, and the ribbons show their standard deviations. Three sets of XGBoost hyperparameters and a fix set of feature engineering hyperparameters are evaluated, $m = 454$, $q = 2.54$, $l = 10$, $account\_CumRain = 1$, $account\_season = 0$, and rainfall depth feature generation option = 1. The hyperparameters values are randomly selected. (b) Comparison of the inner CV errors of the models trained with different sets of XGBoost hyperparameters. The points correspond to the mean values obtained in the inner CV iterations, and the vertical bars show their standard deviations. The feature engineering hyperparameters are fixed and are the same as the figure above.

https://doi.org/10.5194/hess-2020-460
Preprint. Discussion started: 7 October 2020

Hydrology and Earth System Sciences
Discussions
Open Access
EGU

805

Figure 4 (a) Comparison of the inner CV errors of different models built with 100 sets of feature engineering hyperparameters at different outer CV iterations. Each dot corresponds to a model. The lines are fitted using local polynomial regression. XGBoost hyperparameters are optimized for each model. The models are sorted by the mean inner CV error associated with each set of
810 **engineering hyperparameters estimated at the outer CV iterations. Rainfall depth feature generation option 1 is used. (b) The rank of the mean inner CV errors of the models built with different sets of feature engineering hyperparameters at different outer CV iterations. Each dot corresponds to a model, and the models of the hyperparameters are connected by a vertical line.**

https://doi.org/10.5194/hess-2020-460
Preprint. Discussion started: 7 October 2020

Hydrology and Earth System Sciences
Discussions

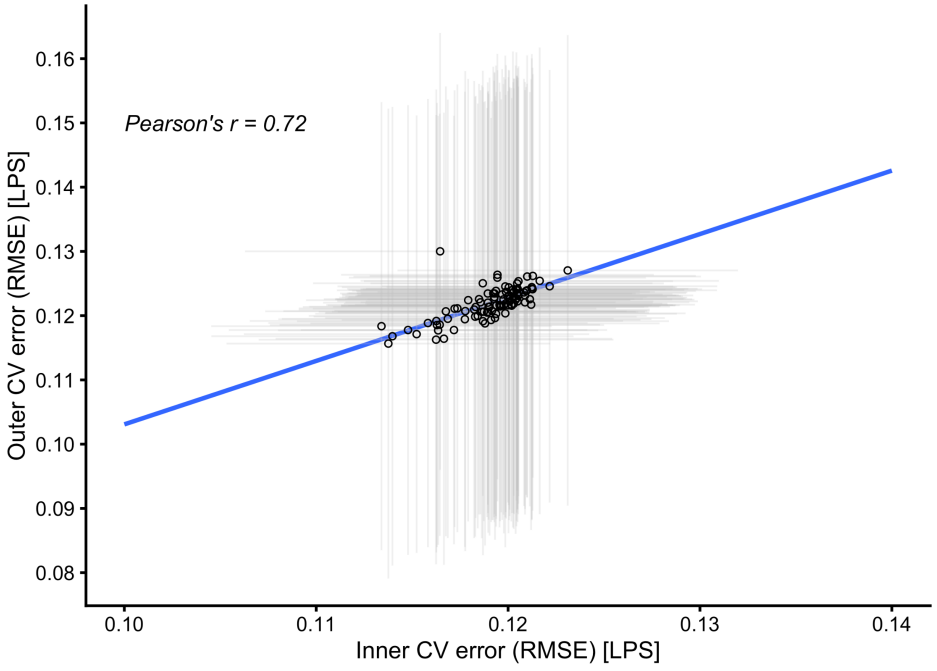


Figure 5 The mean inner CV error compared with the mean outer CV error of different models. Each dot corresponds to models built with a set of feature engineering hyperparameters and the associated optimal XGBoost hyperparameters at different outer CV iterations. The vertical and horizontal lines correspond to the standard deviations of the estimated errors at different outer CV iterations. Rainfall depth feature generation option 1 is used.

https://doi.org/10.5194/hess-2020-460
Preprint. Discussion started: 7 October 2020

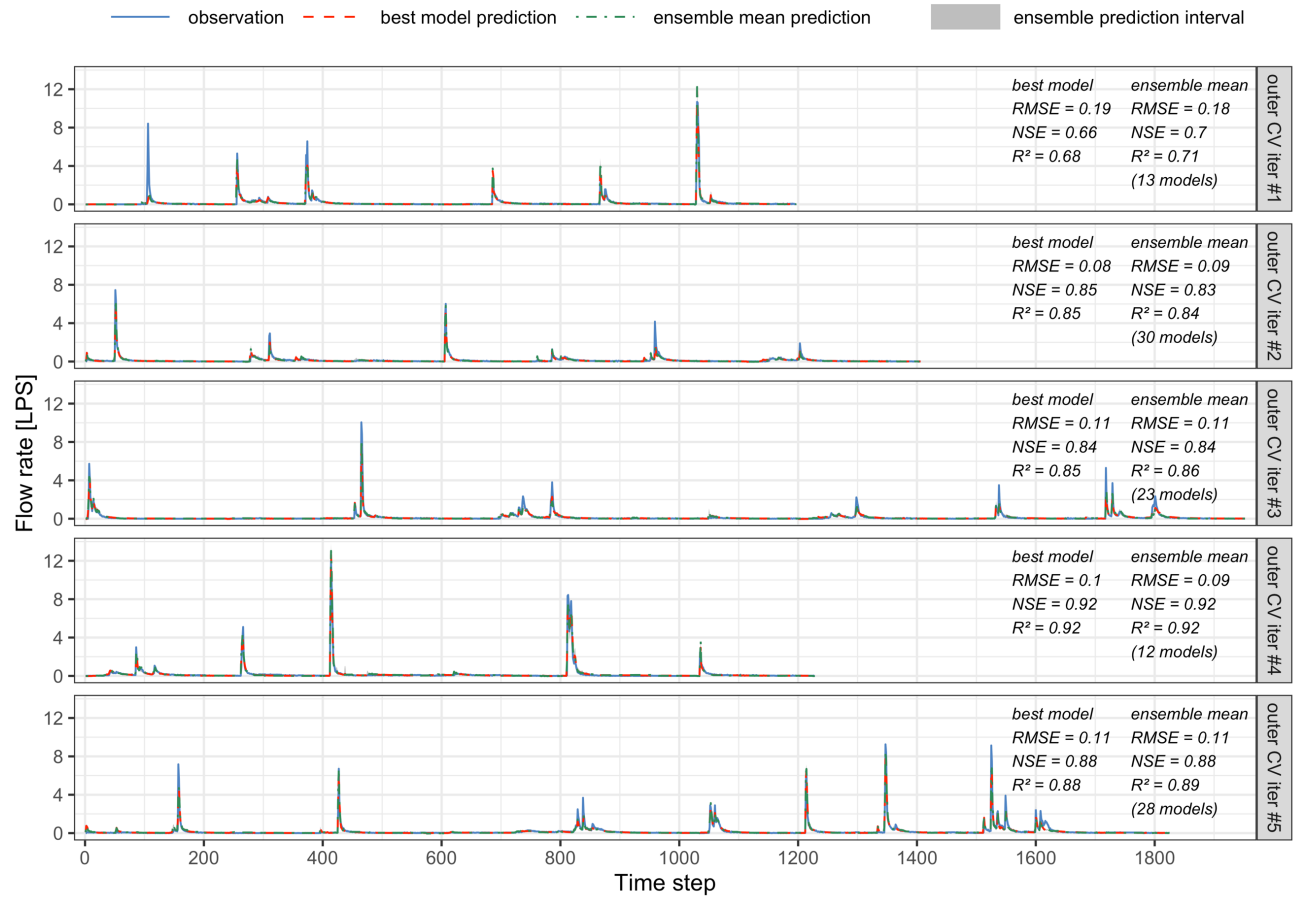
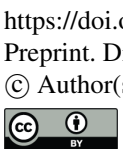


Figure 6 Predicted hydrographs of different models compared with observed hydrographs in the test fold at each outer CV iteration. Each time step is 10 minutes. For clarity, each subfigure shows the five largest runoff events, and the dry periods between the events are not shown. The (very small) shaded areas show the minimal and maximum predictions of the non-dominated models in the ensemble. The performance metrics are shown and are calculated for all the runoff events (including the smaller events that are not shown in this figure). Rainfall depth feature generation option 1 is used.

https://doi.org/10.5194/hess-2020-460
Preprint. Discussion started: 7 October 2020



Hydrology and Earth System Sciences
Discussions
Open Access
EGU

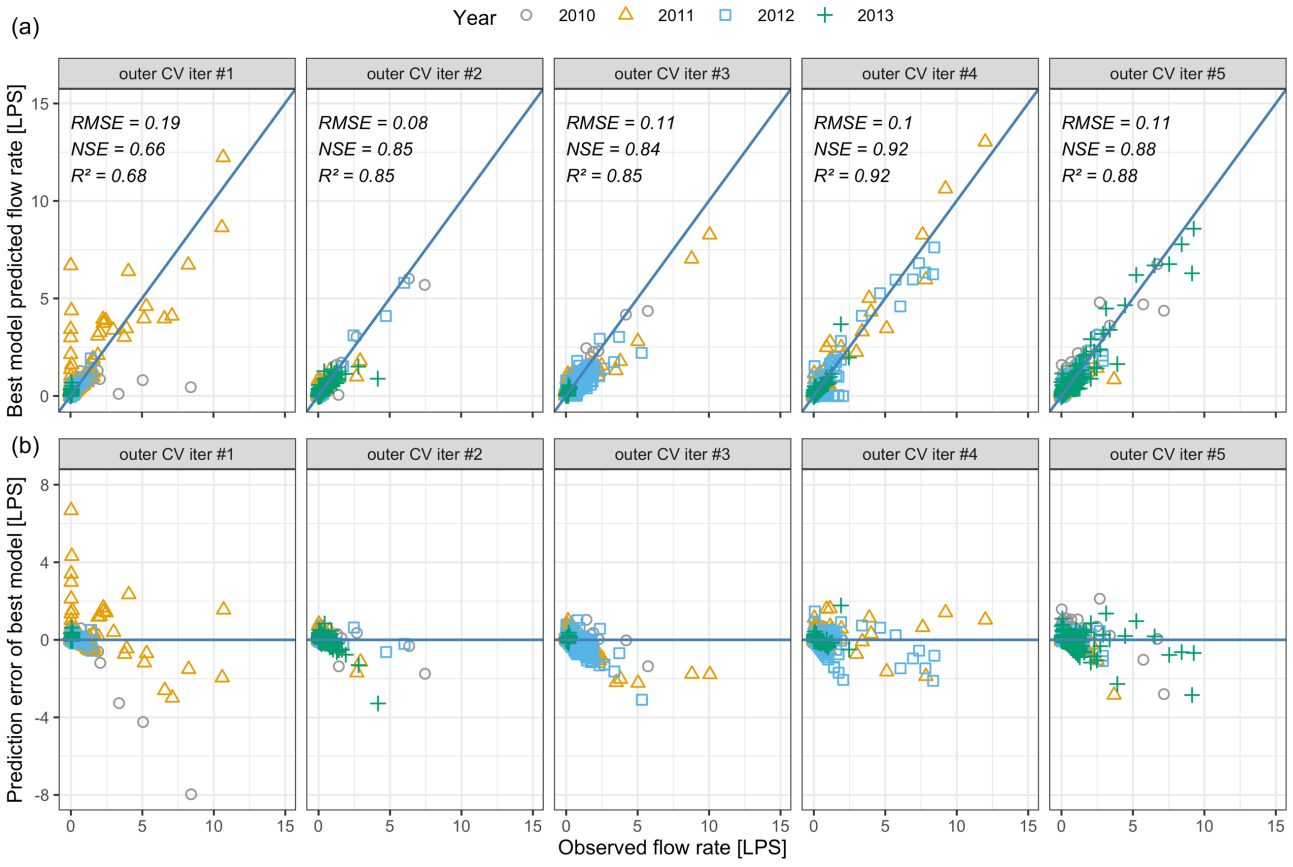


Figure 7 (a) Comparison between the observed flow rates and the predicted flow rates of the best models for the test fold at each outer CV iteration. The performance metrics of the models are shown. Lines of slope of 1 are plotted for reference. (b) Prediction errors of the best models plotted against observed values for each outer CV iteration. Rainfall depth feature generation option 1 is used.

https://doi.org/10.5194/hess-2020-460
Preprint. Discussion started: 7 October 2020

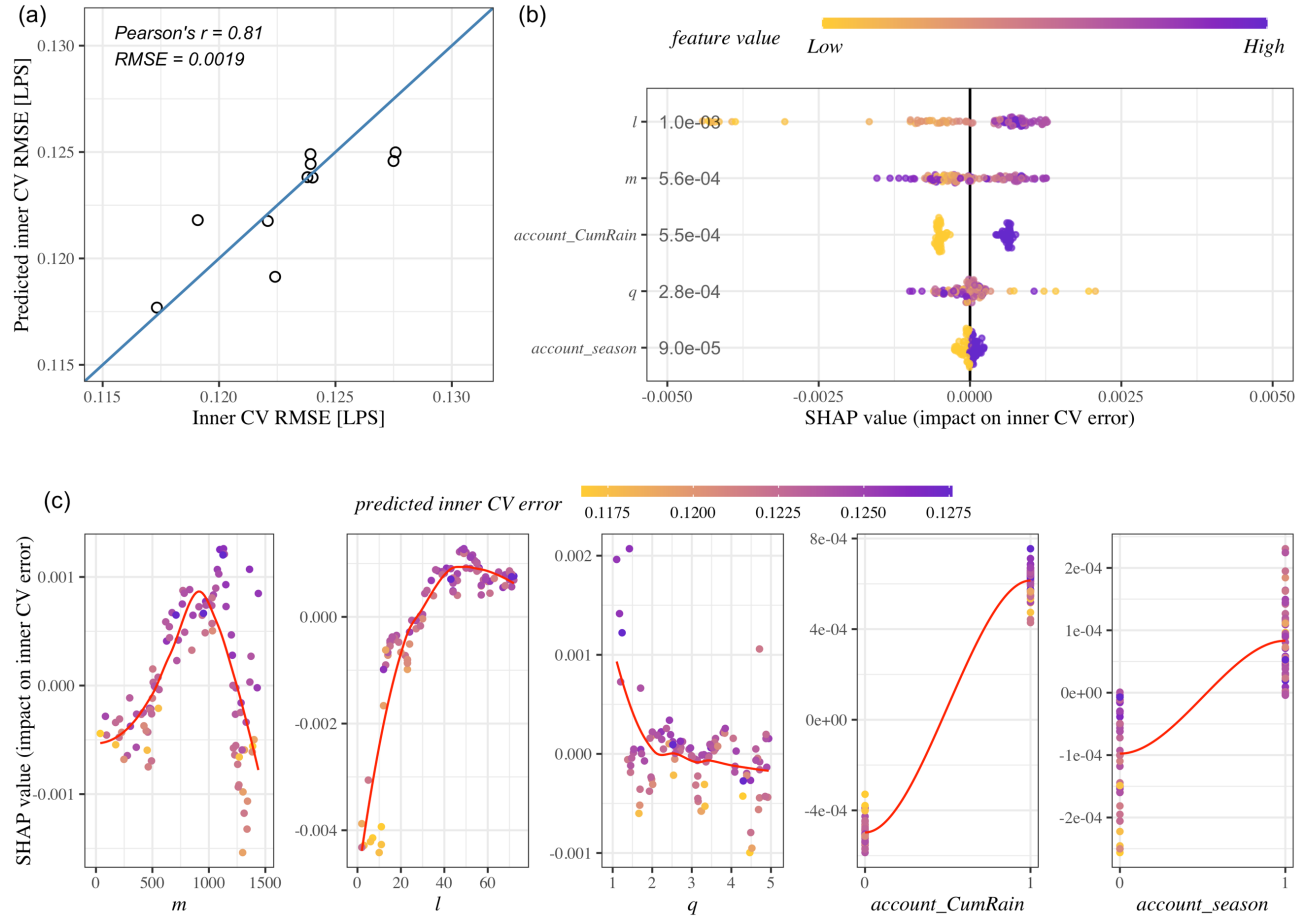


Figure 8 (a) The inner CV RMSE predicted by the XGBoost model compared with the observed inner CV RMSE in the test fold. A line of slope of 1 is shown for reference. (b) SHAP summary plot. Each dot corresponds to a sample, and its x position shows the impact a feature has on the predicted inner CV RMSE. The y positions of some dots are shifted to minimize overlap and to show the density. The numbers on the y-axis show the average SHAP value of the feature and are used for sorting them. The color represents the value of each feature. (c) SHAP dependence plots for each feature. The feature values are plotted against the SHAP values. Each dot corresponds to a sample, and the color represents the predicted inner CV RMSE. The lines are fitted using local polynomial regression. Results for the model of outer CV iteration #3 is shown. Rainfall depth feature generation option 1 is used.

https://doi.org/10.5194/hess-2020-460
Preprint. Discussion started: 7 October 2020

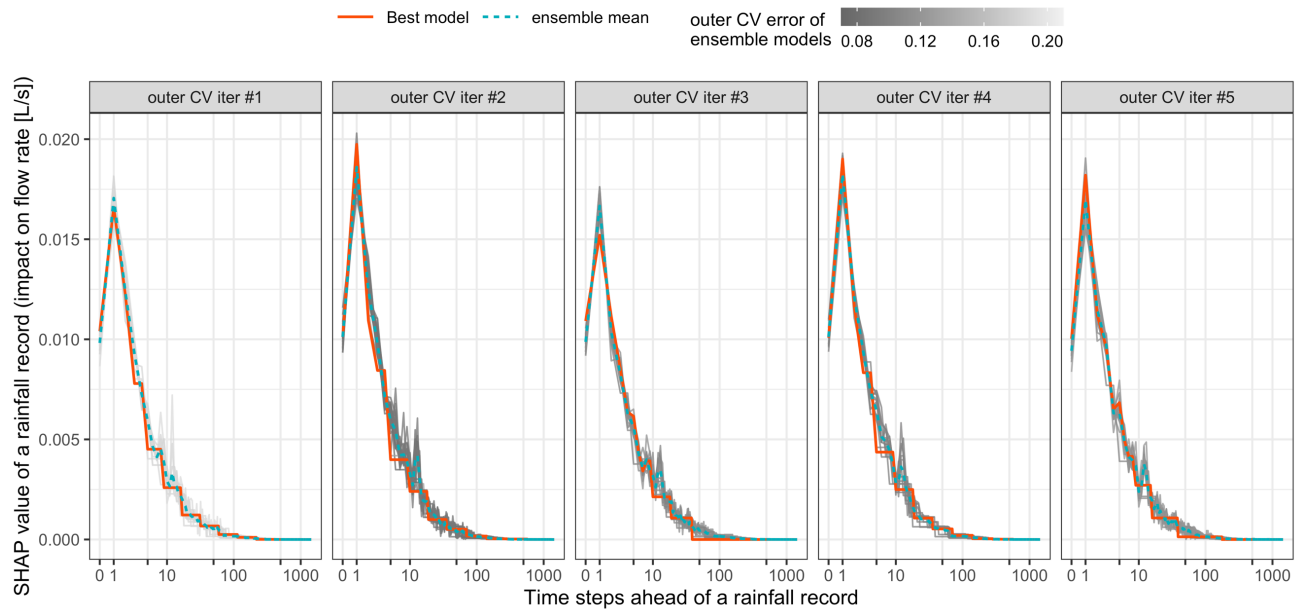


Figure 9 The average contribution of a rainfall depth recorded at a single time step to the runoffs at different time steps ahead. Each time step is 10 minutes. The intensity of the color of the ensemble models indicates their outer CV error (RMSE). The x-axis is in pseudo-logarithm scale. Each subfigure shows the results for an outer CV iteration. Rainfall depth feature generation option 1 is used.

https://doi.org/10.5194/hess-2020-460
Preprint. Discussion started: 7 October 2020

Hydrology and Earth System Sciences
Discussions

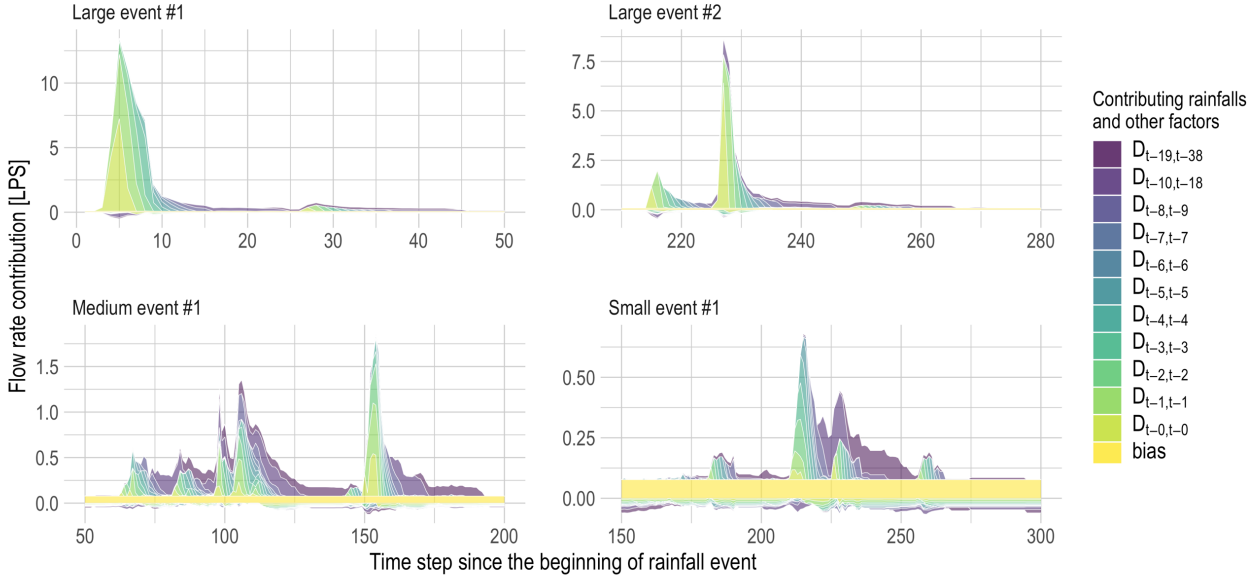


Figure 10 Flow rate contributions of the rainfalls recorded in different time periods and other factors. Each time step is 10 minutes. $D_{t-i,t-j}$ corresponds to rainfall depth recorded between time $t-j$ and $t-i$. Note the flow rate prediction at each time step is the sum of the positive contributions minus the sum of the quantity of the negative contributions (if any). The bias is a constant value and is the prediction made by the model when the values of the input features are unknown. The period corresponds to peak flow is shown in each subfigure. The prediction basis of the best model in outer CV iteration #3 is examined here. Rainfall depth feature generation option 1 is used.

https://doi.org/10.5194/hess-2020-460
Preprint. Discussion started: 7 October 2020

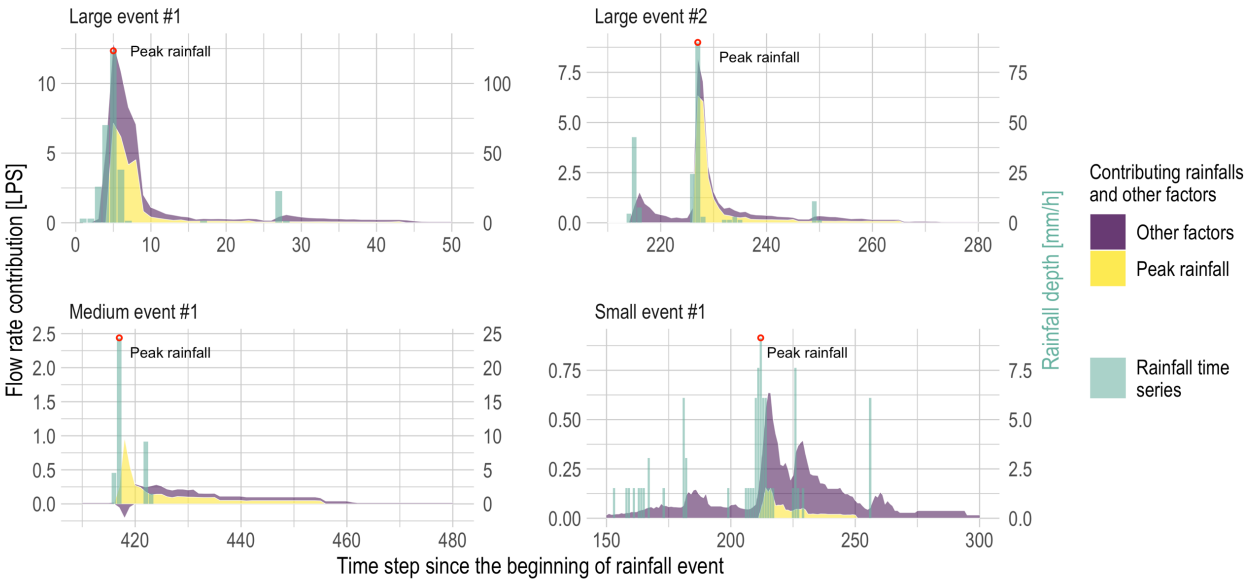
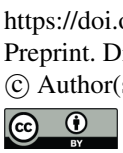


Figure 11 The continuing impact of peak rainfalls on the subsequent runoff events. The contributions of other factors are aggregated for clarity. Each time step is 10 minutes. The hyetograph is shown with the scale in the left y-axis. The period corresponds to peak rainfall is shown in each subfigure. The prediction basis of the best model in outer CV iteration #3 is examined here. Rainfall depth feature generation option 1 is used.

https://doi.org/10.5194/hess-2020-460
Preprint. Discussion started: 7 October 2020



Hydrology and Earth System Sciences
Discussions
Open Access
EGU

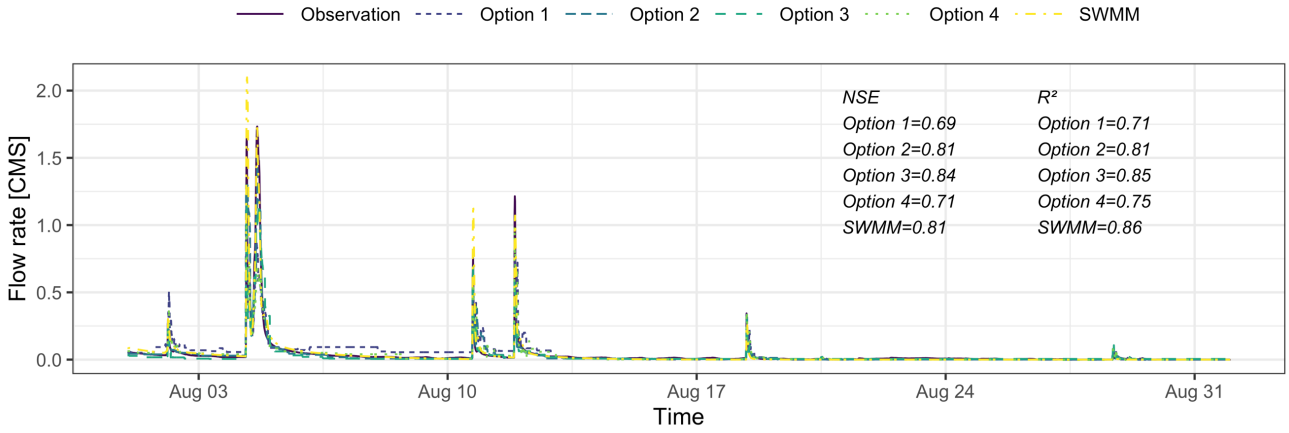


Figure 12 Hydrographs predicted by different models compared with the observed hydrograph for the test period. Options 1 through 4 correspond to XGBoost models built with different rainfall depth feature generation options. RMSE values are listed in Table 3. The SWMM model developed in Lee et al. (2018a) is used.

https://doi.org/10.5194/hess-2020-460
Preprint. Discussion started: 7 October 2020

Hydrology and Earth System Sciences
Discussions

Table 1 Feature engineering hyperparameters and XGBoost hyperparameters.

Hyperparameters	***Values***	***Explanations***
	Feature engineering hyperparameters	
m	A random integer between 12 and 1440	The value corresponds to a look back period ranging from the past 2 hours to the past 10 days, the rainfalls recorded during which are used for runoff prediction.
l	A random integer between 1 and 72	The value corresponds to a period ranging between 10 minutes to 12 hours, the rainfalls recorded during which are considered to be recent and of higher relevance for runoff prediction.
q	A random number between 1.1 and 5	q is the common ratio of the sequence of the distances between the cut points, which form the time intervals for calculating rainfall depth features.
$account\_CumRain$	0 or 1	0 means the cumulative rainfall depths recorded since the beginning of rainfall observation is not used as a feature, and 1 means it is used as a feature.
$account\_season$	0 or 1	0 means the rainfall event occurring month is not used as a feature (using one-hot encoding), and 1 means it is used as a feature.
	XGBoost hyperparameters	
eta	0.02, 0.05, 0.1, or 0.3	Learning rate.
max_depth	2, 4, or 6	Maximum depth of the trees.
colsample_bytree	0.7 or 1	The ratio of the features randomly selected for building each tree.
subsample	0.7 or 1	The ratio of the training samples randomly selected building each tree.
nrounds	2500	The maximum number of boosting iterations allowed in inner CV.
early_stopping_rounds	10	Model training stops if the inner CV error is not improved for 10 iterations. The optimal number of boosting iterations is then used for training models on the entire training (outer) fold.
monotone_constraints	A binary vector of length of the number of features. Each element corresponds to a feature: 1 is used for rainfall depth features, and 0 is used for the other features.	1 means increasing constraint, i.e., there is a monotonic increasing relationship between a rainfall depth feature and the flow rate. 0 means the there is no monotonic constraints.

https://doi.org/10.5194/hess-2020-460
Preprint. Discussion started: 7 October 2020

Hydrology and Earth System Sciences
Discussions

Other hyperparameters, including *min_child_weight, γ, and λ*	Default values used in the XGBoost	A list of the parameters and their default values can be found in the documentation of the XGBoost software (Chen and He, 2020).

870

https://doi.org/10.5194/hess-2020-460
Preprint. Discussion started: 7 October 2020

Hydrology and Earth System Sciences
Discussions

Table 2 Comparison of the prediction accuracy of the models built for the Washington Street (WS) site. The hyperparameters correspond to the smallest mean inner CV error are chosen during each outer CV iteration. The models trained on the training folds (outer) with the optimal hyperparameters are evaluated using the corresponding test folds to obtain the outer CV RMSE. The mean values across different outer CV folds are shown, and the standard deviations are given in the following brackets.

Model	Inner CV RMSE (standard deviation) [LPS]	Outer CV RMSE (standard deviation) [LPS]
XGBoost - option 1	0.112(0.009)	0.119(0.039)
XGBoost - option 2	0.098(0.013)	0.104(0.041)
XGBoost - option 3	0.102(0.011)	0.108(0.039)
XGBoost - option 4	0.110(0.010)	0.113(0.040)
Linear regression - option 1	0.161(0.006)	0.163(0.024)
Linear regression - option 2	0.161(0.006)	0.163(0.024)
Linear regression - option 3	0.161(0.006)	0.163(0.024)
Linear regression - option 4	0.161(0.006)	0.163(0.024)

https://doi.org/10.5194/hess-2020-460
Preprint. Discussion started: 7 October 2020

Hydrology and Earth System Sciences
Discussions

880 **Table 3 Comparison of the prediction accuracy for the models built for Shayler Crossing Watershed (SHC). The SWMM model developed in Lee et al. (2018a) is used, where it was calibrated on both the training and the test folds used in this study. The linear models built with different feature generation options are nearly identical, thus their performances are shown in the same row.**

Model	RMSE of validation fold [CMS]	RMSE of test fold [CMS]
XGBoost - option 1	0.054	0.071
XGBoost - option 2	0.066	0.055
XGBoost - option 3	0.051	0.052
XGBoost - option 4	0.054	0.069
SWMM	0.101	0.056
Linear regression	0.120	0.080