Comparison of machine learning algorithms for emulation of a gridded hydrological model given spatially explicit inputs

Author Names and Affiliations

Theodore Lim, School of Public and International Affairs, Virginia Polytechnic Institute and State University (**corresponding author**). 140 Otey St NW, Rm 207, Blacksburg, VA USA 24060
tclim@vt.edu

Kaidi Wang, School of Business, Macau University of Science and Technology, kdwang@must.edu.mo

Code Availability

ParFlow.CLM is available at: https://github.com/parflow
Code developed for producing training data for this study is available at: https://github.com/theochli/parflow_py_utils

Authorship Statement

Theodore Lim contributed to the conception, hydrological simulation, data analysis, and writing of the manuscript. Kaidi Wang contributed to machine learning training, data analysis, and writing of the manuscript.

Abstract

This study compares the performance of several machine learning algorithms in reproducing the spatial and temporal outputs of the process-based, hydrological model, ParFlow.CLM. Emulators or surrogate models are often used to reduce complexity and simulation times of complex models, and have typically been applied to evaluate parameter sensitivity or for model parameter tuning, without explicit treatment of variation resulting from spatially explicit inputs to the model. Here we present a case study in which we evaluate candidate machine learning algorithms for suitability emulating model outputs given spatially explicit inputs. We find that among random forest, gaussian process, k-nearest neighbors, and deep neural networks, the random forest algorithm performs the best on small training sets, is not as sensitive to hyperparameters chosen for the machine learning model, and can be trained quickly. Although deep neural networks were hypothesized to be able to better capture the potential nonlinear interactions in ParFlow.CLM, they also required more training data and much more refined tuning of hyperparameters to achieve the potential benefits of the algorithm.

Keywords
Emulation modeling, surrogate modeling, ParFlow.CLM, machine learning

1 Introduction

Emulation models, also called "meta models" or "surrogate models," are models of models that are intended to reduce the computational resources or complexity of the original model in order to facilitate faster simulations. They are often used to optimize or calibrate parameters of the original model (Ratto, Castelletti, and Pagano 2012). In such applications, the spatial arrangement of the domain remains constant (for example, the locations of various land covers); varied from iteration to iteration of the model are the parameter values, usually according to predefined probability distributions meant to represent the uncertainty around the parameter values. In many cases however, it is not only the values of input parameters that require sensitivity testing; it is also necessary to understand how the spatial arrangements and configurations of parameters contribute to variability. This is especially true in spatial decision-making, where stakeholders are interested in the potential effects of various spatial arrangements corresponding to future alternatives. For testing spatial "what if" scenarios, it is primarily the spatial arrangement of zones within the domain that is changing between model iterations, not the values assigned to parameters (Klosterman 1999). The potential application of emulation models to capture spatially resolved inputs and outputs however remains under-explored in the literature.

In this research we provide a simplified case example that will contribute to development of a broader characterization of original model complexities to be emulated by ML algorithms. We use the case to illustrate criteria that may be used in practical contexts where emulators need to capture spatially resolved inputs and preserve spatially and temporally resolved outputs. In addition to fidelity to the original model outputs, the criteria also include: (1) total time savings; (2) simplicity of the training process; and (3) robustness of the emulator to the algorithm's hyperparameters and new scenarios.

In the following section, we present a review of related research on the use of ML in hydrologic modeling, emulation, and relevance for spatial scenario testing.

2 Related Research

2.1 Model emulators and ML algorithms in physical hydrology simulation

An emulation model (also known as a "meta-model" or "surrogate model") is a model that mimics the outputs of an original model (Forrester, Sobester, and Keane 2008; Ratto, Castelletti, and Pagano 2012; Razavi, Tolson, and Burn 2012; Asher et al. 2015). There are two major approaches to emulation modeling: response surface emulators and lower fidelity surrogates (Razavi, Tolson, and Burn 2012). The primary purpose of the former is simply to reproduce the output of the original model with high fidelity. The "response surface" is the original model's output, and the closer the surrogate can come to reproducing the surface under new input parameter values, the better. In the response surface emulator approach, the processes included in the model are less important than in lower fidelity surrogates. In this latter approach, the physical bases of the model's structure are preserved and computational cost savings are instead achieved through eliminating less relevant detail in the original model (e.g. Haasnoot et al. 2014). Compared to lower fidelity surrogates, response surface emulators are better suited for replacement with machine learning-based algorithms, which are often "black boxed" processes that can result in high fidelity output, but do not preserve physical meaning of processes.

Machine learning (ML) refers to the use of statistical algorithms and sampling techniques to automatically extract patterns from data, and has been used in emulation processes. They have often been applied in modeling meta-analyses, such as in evaluating model parameter sensitivity, parameter optimization or systems modeling, which require iterative sampling and many model realizations under different parameters or starting conditions (Pianosi et al. 2016). They have also been used to speed scenario testing (Carnevale et al. 2012), and increasingly suggested in integrated assessment models to reduce computational barriers in coupling between sub-models for various scales, and diverse social and environmental contexts and levels of complexity (Little et al. 2019; Liu et al. 2008; Mahmoud et al. 2009).

Recently in the hydrological sciences, ML has been applied as response surface emulators to relate high-dimensional input and output fields as images (Mo et al. 2019); to reconstruct the shapes of unit hydrographs of surface runoff for any combination of input parameters (Moreno-Rodenas et al. 2018); to evaluate uncertainty of a spatially semi-distributed hydrologic model (Yang et al. 2018); to optimize water resource management decisions including several interconnected sub-models (B. Wu et al. 2015); and to predict hydrologic connectivity metrics over spatially heterogeneous land covers (Crompton, Sytsma, and Thompson 2019). In these examples, ML algorithms are applied to the inputs and outputs of physical simulation models in order to speed calculations and ease computational requirements. Physics or process-based simulation models usually require solving partial differential equations sequentially for multiple time steps, which can have a high computational load (e.g. Shen et al. 2016). To handle these loads, nonlinear algebraic solvers have been developed to distribute computational resources over parallel, or multi-threaded processes (Hindmarsh and Taylor 1998).

Deep learning (DL) algorithms comprise a subset of ML algorithms that are based on artificial neural networks, structures that relate inputs and outputs of statistical models to each other through layers of connecting "neurons" or cells. The "deep" in DL refers to the stacking of many layers within the network. Recently, DL has driven rapid advances in artificial intelligence, including in computer vision and speech recognition, in which highly complex, nonlinear data patterns need to be captured. For a review of DL and its uses and potential applications in hydrologic modeling, the reader is referred to (Shen 2018). With respect to hydrologic simulation, Shen identifies DL as a potential opportunity for addressing challenges with model scaling and equifinality, especially with respect to fully resolving dynamics (such as those represented by Richards equation) at high resolutions. Others have also addressed the potential for DL algorithms to increase computational performance solving the partial differential equations themselves (Han, Jentzen, and E 2018).

2.2 Spatial considerations of environmental decision-making and the need for multi-level fidelity

For land management decision-support processes such as those in which urban or regional planners may be involved, users are often interested in if and how spatial arrangements of alternatives will influence model outputs relative to other alternatives. This is a particular need in circumstances where high levels of spatial heterogeneity could result in very different hydrologic responses predicted by models than if the spatial heterogeneity was not represented. In addition, whether or not such complexity in a model will actually result in a practically significant change in output requires evaluating both sensitivity of the model to different parameterizations and scenarios. The process of sensitivity analysis and model complexity determination should be a part of an iterative modeling process, in which evaluation of the relevance of including spatial heterogeneity at different resolutions is evaluated (Jakeman, Letcher, and Norton 2006).

In practice however, the selection of a model of a given complexity is usually subject to factors that are neither scientific nor explicitly evaluated according to the goals of the environmental decision-making context. Some models that have been widely adopted remain based on simple heuristics that do not reflect the specifics of the site. For example, for determining how much urban development should occur within a given watershed, many urban and regional planners rely on the “10 percent rule” as a motivation for clustered development and limiting sprawl (Berke et al. 2003; Schueler, Fraley-McNeal, and Cappiella 2009; H. Wu et al. 2015). This rule was based on early empirical studies that found that statistical differences in streamflow signatures could be detected in watersheds with as little as 10 percent of the land surface area converted to impervious surface (Schueler 1994). Continued reliance on this heuristic persists, despite subsequent improvements in scientific understanding that imperviousness at lower thresholds can result in detectable degradation (Booth and Jackson 1997); the position of development relative to the topographic drainage networks was also shown to be important, where upslope imperviousness was less impactful than downslope imperviousness (Moglen and Kim 2007); and that it may be more the connectivity of infrastructure and land cover change in general, rather than total impervious area that are associated with largest impacts to hydrology (Alberti and Booth 2007; Smith and Smith 2015; T. Lim 2016). In environmental studies more generally, model simplification in environmental decision-making contexts is attractive because simpler models are often more generalizable, can better summarize or represent causal relationships in an understandable way, are often more generalizable (but less precise), and have lower computational requirements than more complex models (Hong et al. 2017).

Models may also tend toward unwarranted levels of complication for non-scientific reasons. These reasons include: "showoff factor," "include all" syndrome, and "possibility factor" (Chwif, Barretto, and Paul 2000). At times, more complex models are perceived to be "more scientific" than simple models (Oreskes 2003). In others, the perceptions, values, and interests of stakeholders could influence decisions about the model. In the Chesapeake Bay Watershed in the United States, for which regulations and policies are closely based on the outputs of an integrated model that includes a hydrological model, stakeholders often pushed for the inclusion of more detail in the model in order to better represent processes important from their perspectives. For example, stakeholders pushed to include high levels of spatial heterogeneity in land use/land cover and highly specific nutrient loading submodules representing farmer behavior were important for stakeholder buy-in of the model and inductive learning from scenario testing (T. C. Lim 2021). In these examples, dissatisfaction with computational resources and simulation time may be proffered as reasons for decreased credibility of models that may occlude non-technical or political reasons for delaying the acceptance of a given environmental model.

The above examples illustrate why lowered computational thresholds for testing the effects of spatial resolutions in environmental decision-making contexts are necessary. On the one extreme, a heuristic is the most simplistic "model" requiring the least amount of computation. On the other extreme, a multitude of stakeholder-dependent, spatially-explicit considerations could drive model complexity to the point of unmanageable computational requirements. Based on this, the use of ML algorithms in emulation modeling may be one approach to enabling rapid testing of whether and how the gap between simplistic heuristics and the latest first-principles-based scientific representations of system dynamics may be bridged.

Simplified Heuristic	**Complex Model**
Positive Features ● Quick, convenient ● Immediately deployable ● Easily comprehensible ● Possibly more accurate	Positive Features ● More comprehensive ● High level of specificity ● Incorporates more sources of knowledge (scientific and stakeholder) ● Can accommodate higher levels of spatial heterogeneity ● Possibly more precise
Negative Features ● Not spatial ● Too simplistic ● Not precise ● Would not reflect differences between spatial alternatives	Negative Features ● Risk of overcomplication ● Potentially less accurate ● Difficult to deploy, validate ● Long run times, high computational cost

The motivation for this study is therefore to evaluate the performance of several ML algorithms' performance in emulating the high spatial and temporal resolution outputs of an original model that accepts spatially explicit inputs. Given the motivation of rapid testing for the purpose of determining appropriate levels of model complexity for decision-making contexts, we evaluate several algorithms' emulator output fidelity, complexity of training, and training data requirements to improve understanding of what kinds of high resolution, complex hydrologic model outputs can be represented by what kinds of ML algorithms.

3 Methods

The comparison of several ML algorithms performance in emulating a high resolution, coupled-surface subsurface hydrological model consisted of four main steps: (1) Simulating a hypothetical domain, with spatial permutations of hydrologic parameters (scenarios) using ParFlow.CLM (Maxwell and Miller 2005); (2) Summarizing the three-dimensional, temporal outputs into both one-dimensional temporal outputs (hydrographs) and two-dimensional spatial outputs (raster arrays) for each scenario; (3) Using the outputs corresponding to the scenarios to train emulators for the temporal and spatial outputs using several ML algorithms (**Figure 1**).

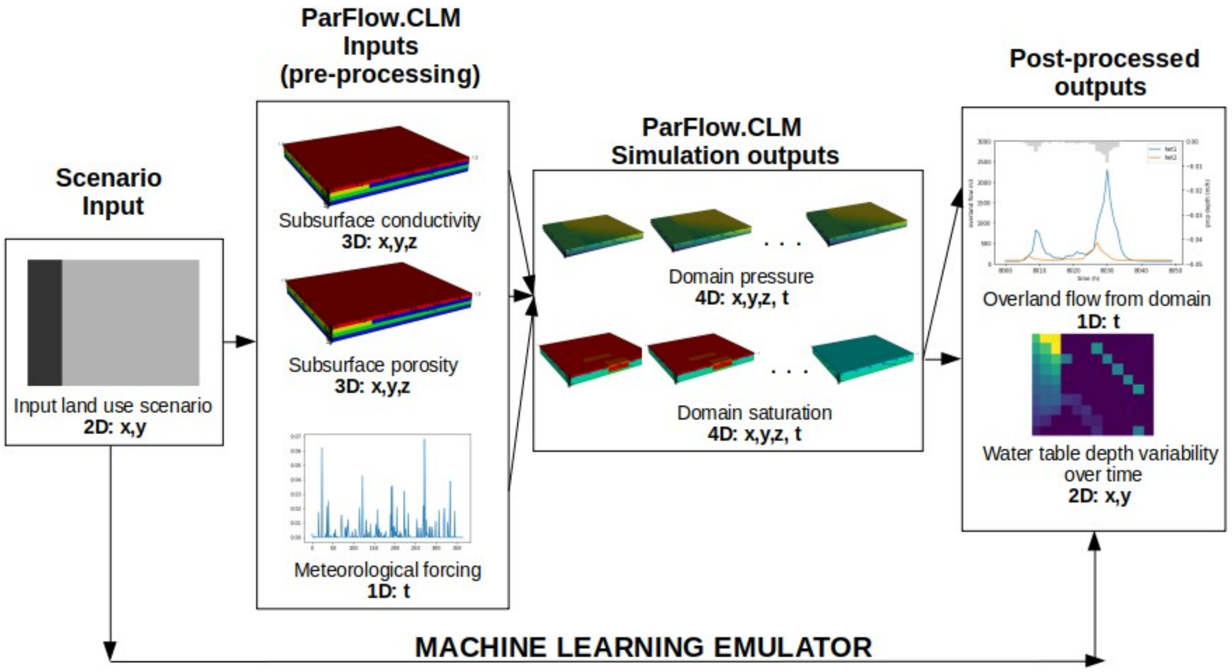


Figure 1. Overview of method generating training samples using ParFlow.CLM, post-processing, and emulation using machine learning emulators.

3.1 Description and Motivation for Case Study: ParFlow.CLM Gridded Hydrologic Model

The case chosen to illustrate spatial temporal emulation using spatially-resolved inputs is a hypothetical domain with varying magnitude and placement of an "urban patch" within a forested hillslope.

Process-based models are particularly useful for capturing "non-linear" behavior, such as variable source area, which results from interactions between surface and groundwater interactions. ParFlow.CLM is such a three-dimensional, distributed hydrological model, able to represent surface-subsurface interactions that could result in nonlinear runoff responses. ParFlow.CLM is based on the Richards equation and solves differential equations for each cell in the three-dimensional domain for each timestep of the simulation. This makes ParFlow.CLM as a model highly generalizable, but also highly computationally expensive, and therefore a candidate for emulation using ML. The spatial domain used in this study was specified to exhibit these interactions, using the "slab" case (Maxwell et al. 2014), which has been used to demonstrate differences in surface-subsurface interactions and runoff generation in a range of integrated hydrological models.

We specified a simple hypothetical domain that we expected to exhibit some evidence of non-linear dynamics in runoff generation processes that would be uniquely captured by an integrated surface-subsurface hydrological model. The domain specification described here is similar to the "slab" case used in Maxwell et al., (Maxwell et al. 2014), which was used to demonstrate differences in surface-subsurface interactions and runoff generation between a range of integrated hydrological models. The "slab" case shows how subsurface heterogeneity affects runoff process, with low conductivity areas primarily controlled by infiltration-excess processes, and higher conductivity only generating runoff through saturation-excess. With this formulation, different integrated model formulations have been shown to exhibit significant differences in both timing and peaks of runoff processes from the domain, due to differences in spatial extent of saturated areas (Maxwell et al., 2014). For the range of spatial scenarios that we were interested in here, this was of particular interest.

The dimensions of the hypothetical native domain grid were 10 x 12, with each grid cell's resolution 100m x 100m in the horizontal plane, and variable dz thicknesses (from bottom to top of domain: 2m, 2m, 2m, 1m, 1m, 1m, 0.5m, 0.5m, 0.5m, 0.5m), for a total domain depth of 11m. The hypothetical domain was designed to include two hypothetical land cover types: forested and urban, to which we assigned different subsurface hydrological parameters-- porosity, and saturated hydraulic conductivity-- and land surface properties in the top-most four layers. Land surface parameters were held at the defaults for the CLM portion of the coupled ParFlow.CLM model, with "broadleaf forest" type assigned to the forest land cover, and "urban/bare" type assigned to the urban land covers (Oleson et al. 2010). The hydrologic parameters used in the simulations can be found in the Appendix. The slopes in the x and y directions were both set to 1%. The terrain-following grid option in ParFlow was used. All scenarios were forced using 1D meteorological forcing from NLDAS historical data product for a location in central Pennsylvania, representing a humid climate (Collick et al. 2015). The year 2010 was used for all scenarios for both the spinup process and for the one-year simulation. 2010 was chosen because this year had a total annual precipitation depth closest to the 100-year mean (1901 – 2000) in the past 10 years, with few monthly precipitation total outliers compared to the 100-year monthly means. The spinup process was implemented in a four step process, described in more detail in Lim and Welty, 2017. Spinup was carried out for all scenarios until dynamic equilibrium was observed.

Given the size of the grid cells in this case, we conceptualized hydraulic conductivity and porosity parameters of the "urban" land use as aggregated upscaled values for the average of the covered area (Bhaskar et al. 2015). Because ParFlow.CLM is a hydrologic model, there is not a straightforward way to represent hydraulics within drainage infrastructure systems that are known to play a dominant role in the fate of urban runoff. Others have represented subsurface infrastructure networks in hydrologic models before (Bhaskar et al. 2015; Barnes, Welty, and Miller 2018; Voter and Loheide 2018; T. C. Lim and Welty 2017), however, for the purposes of this study, which is focused on understanding how ML algorithms can preserve spatial explicitness, we chose not to introduce this additional layer of complexity.

To generate scenarios, we created contiguous, rectangular patches of "urban" land cover in the domain, for between 10 and 30 grid cells of the 120 grid cell total, for a total urban area between 100,000 m2 ~ 300,000 m2 in the domain. We created all rectangular dimension/orientation permutations of this area that would fit fully within the domain and systematically moved the position of the urban patch throughout the domain. This resulted in 444 spatial scenarios that were then used to distribute the values of hydrologic parameters in Table 1 for "urban" and "forested" land uses. The spatial inputs were used to run the fourth stage of the spinup (above) was run, and after dynamic equilibrium was verified, the spun-up domain was used to simulate one year of coupled surface-subsurface hydrology from the site. The model ran at 1-hour timesteps for one year, for a total of 8760 three-dimensional array outputs (12x10x10 native grid) for each scenario. Simulations were carried out using four out of 36 cores of a 36-core server. Each scenario required 2 – 3 hours of wall-clock time to complete.

3.3 Machine learning model tuning and selection

We trained ML models for the prediction of hourly streamflow and the variability of groundwater depth separately. The ML algorithms used in this study include Deep Neural Network (DNN), K-nearest neighbor (KNN), Gaussian Process Regressor (GPR), and Random Forest Regressor (RFR). These algorithms were selected to represent a range of ML approaches of varying complexity, with the DNN (a multiplayer perceptron) in particular was included for its hypothesized capability to represent nonlinearities in the time series and spatial outputs. GPR and RFR are also known as powerful nonlinear regressor methods and simpler to implement than DNN, while KNN is the simplest to implement, and the simplest algorithm to implement.

The models take as input features the 12 x10 grid representation of the spatial distribution of urban and forest grid cells rearranged as a 120 x 1 vector. We tested two prediction outcomes: a time-series of streamflow (hydrograph of 180 time steps, represented as a 180 x 1 vector), and a two-dimensional spatial array of the standard deviation of groundwater depth over the 180-hour period for each horizontal position in the domain (120 cell grids, rearranged into a 120 x 1 vector). The input for each scenario was therefore a 120 x 1 land use vector, and outputs were either a 180 x 1 steamflow vector (henceforth “streamflow output”) or a 120 x 1 spatial groundwater variability vector (henceforth “groundwater output”) (See **Figure 1**).

The candidate ML algorithms (DNN, KNN, GPR, and RFR) were then used to relate inputs to outputs. The pipeline is three-fold. First, we randomly divide all the scenarios into a training set and a test set using an 80/20 split size. The training set is, then, used to tune the models to select the best combination of the hyperparameters for each type of model. This study uses five-fold cross validation to tune the models. Finally, we report the performance of the best tuned models on the test set and evaluate how the model performance varies with training size and stratification.

In the following part of the section, we sequentially introduce the measurement we use to evaluate the fidelity of the emulator to the original model, the model tuning process including the model tuning tool and hyperparameter selection criteria, and the sensitivity analysis of the model performance to the training size and stratification.

3.3.1 Measures of emulator fidelity to original model for each scenario and evaluation of the emulator’s performance across the scenario set

The performance of each trained emulator was evaluated in a two-step process. First, we calculated the fidelity of the output of the emulator to the original ParFlow.CLM post-processed output. For this step, we used commonly-used metrics for model skill, the Nash-Sutcliffe Efficiency (NSE) for streamflow, and R2 for groundwater variability.

NSE is a common metric for summarizing the performance of a model compared to observed data. NSE is calculated for each scenario in the test set as in Equation 1.

$$(1)\, NSE = 1 - \frac{\sum_{t=1}^{180} \left(Q_t - \widehat{Q}_t \right)^2}{\square}$$

Where Q_t is the flow at time = t output from the original model, Q_t(hat) is the emulator’s prediction, and Q_t(bar) is the mean streamflow in the 180-h time series.

The fidelity of the emulator for each scenario’s groundwater variability was evaluated by calculating a spatial coefficient of determination (R^2). R^2 is calculated for each scenario in the test set as in Equation 2.

$$(2)\, 1 - \frac{\sum_{i=1}^{10} \sum_{j=1}^{12} \left(y_{i,j} - \widehat{y_{i,j}} \right)}{\square}$$

Where i and j indicates the x and y position index of each cell in the horizontal domain; y refers to the original model output standard deviation over the 180-h period at position i, j; y(hat) is the emulator-predicted standard deviation of hourly water depth in the year at position i,j; and y(bar) is the mean standard deviation in the domain at position i,j according to the original model.

In the second step of the evaluation of each emulator, we calculated the proportion of the previously unseen scenarios that the emulator was able to predict with *high fidelity*, where we considered NSE > 0.7 and R2 > 0.7 a scenario predicted with high fidelity. We used this two-step evaluation process across a range of training set sizes to test how each of the emulators performed when provided with more or less training data, hypothesizing that more complex emulators would require larger training sets in order to result in higher proportions of high fidelity scenarios in the test set. The two-step evaluation was also used in the process of tuning the hyperparameters of each emulator.

3.3.2 Hyperparameter tuning using five-fold cross validation

There are two goals of hyperparameter tuning: (1) we aim to maximize the model performance in terms of predicting the steamflow output and the groundwater variability of each scenario; (2) it is necessary to ensure the fidelity of the model on an unseen dataset. In other words, we need to ensure it is not overfitted towards specific training data.

We tuned the hyperparameters using five-fold cross validation, where, in each iteration, the model with selected hyperparameters was trained using four folds and tested with the other fold. The combination of the hyperparameters was then evaluated using the average percentage of the scenarios with an NSE (i.e., for steam flow) or R^2 (i.e., for groundwater variability) larger than 0.7, as the goal of the model tuning is to maximize the share of predictions for scenarios with good performance. This process was repeated for all the models. We selected the best hyperparameters for each model, after visiting the combinations within the search space as listed in Table 1 in the Appendix. We used Scikit-learn (https://scikit-learn.org/), a Python library for machine learning, to implement the model training and prediction. Especially for the more complex models, such as DNN, there are more advanced tools that can aid model training and hyperparameter tuning, such as Tensorflow. However, for this application which attempts to evaluate implementability of emulator training for decision support purposes, it was critical for all candidate algorithms to be trained using similar tools to compare performance given similar upfront investments in training. Tensorflow-based training is more complex and requires a more specific setup.

We also used Neural network Intelligence (NNI) (Microsoft Research (MSR) 2021) to optimize the path of the hyperparameters search, as it is time consuming to exhaust all the combinations. In addition to hyperparameter tuning, DNN also requires a decision for the structure of the network. For this, we first manually tuned the number of layers and the number of perceptrons in each layer using the default hyperparameter setting of DNN in Scikit-learn. The selected DNN structure is 10, 10, 10, 10, 10, 90, 90, 90, 90, and 90. This structure was then used for hyperparameter tuning.

Finally, each model with corresponding best selected hyperparameters was trained using the whole training set and evaluated on the hold-out test set. We report the fidelity and performance of each model using the percent of scenarios with good model prediction performance (i.e., NSE > 0.7 or $R^2 > 0.7$) on the test set.

3.3.3 Evaluation of Emulator Complexities: Robustness to Training Sample Size

Because the candidate algorithms have varying levels of complexity, it is also necessary to assess the training data needs of each algorithm in order to achieve good performance.

Generally, a more complex algorithm (such as DNN) may have greater potential to produce high fidelity outputs, but will also require more training data and have higher hyperparameter tuning requirements. An emulator should not only have high fidelity to the original model, it also needs to be able to reduce overall computation time and exhibit stability in performance when there is limited data for model tuning and training. Therefore, in this study, we explore the potential of the ML algorithms in reducing computation time from two perspectives: (1) we investigate how the performance of various ML models vary over different training sizes and identify the model that generates fidelity even when the training size is small; (2) we test whether a sampling method, i.e., stratified sampling, can improve the model performance while mitigating the requirement for the size of model input.

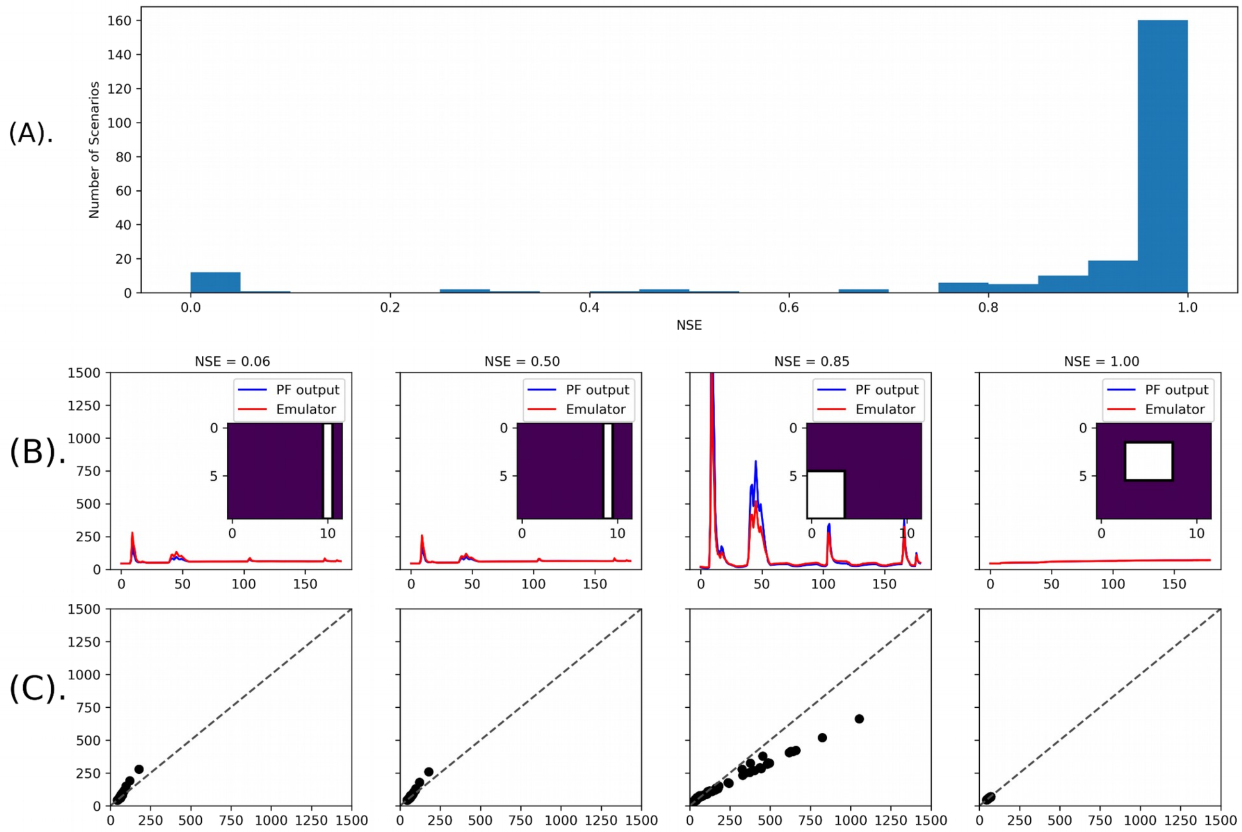
To compare the trade-offs between prediction accuracy and the simulation cost of preparing data for machine learning models, we test four train/test splits: 10%-90%,40%-60%,50%-50%, and 80%-20%. Typically, ML models generate higher prediction accuracy on unseen test scenarios when they are trained with more data, if sufficiently tuned. Therefore, we expect the model performance to increase with the training size.

Stratified sampling is another way that may improve the amount of information presented to the ML algorithms when the training size is limited. We tried several stratification sampling procedures to ensure that the most variability in spatial scenarios were incorporated into the training data, including: stratifying by position of the urban land use relative to the outlet (distance from centroid to the outlet, and distance from the closest corner of the urban land use to the outlet), and stratyding by the size of the urban patch. The aim of stratification is to produce a training data set that is representative of the variation in the overall population of scenarios.

4. Results

4.1 Comparison of model performance by ML algorithm on test set

Across the range of training set proportions, and across the emulators based on the four ML algorithms, there was always a range of fidelities that could be achieved by the fitted emulator on the scenarios in the test set. To illustrate this, below we show the performance of one ML algorithm (RFR) on the scenarios in the test set of a 50%-50% train-test split of the data. **Figure 2A** shows the distribution of NSEs for streamflow. **Figure 2b** shows the original model output 180-h streamflow time series in blue, compared with the emulator output in red, for four scenarios in the test set. These four scenarios represent a range of fidelities (NSE = 0.06, 0.50, 0.85 and 1.00, respectively) to illustrate the emulator's range of performance. The scenarios that had elongated urban patches were predicted with less fidelity than scenarios where the urban patches were more compact. **Figure 2C** shows scatterplots of the hourly streamflow predictions from the original model compared with the emulator, and shows that the lower NSEs were attributed to overprediction of streamflows by the emulator compared to the original model, while for scenarios with higher fidelities (higher NSE), the emulators slightly underpredicted streamflows compared to the original model.



Figures 2A~C. 2A) Distribution of NSEs achieved by the RF-based emulator on 222 test set scenarios. **2B)** Comparison of original model streamflow output and emulator streamflow output, shown alongside input spatial scenarios. White block indicates the "urban patch" within the forested domain. **2C)** Scatterplot of each hourly streamflow output vs emulator prediction. Points above the dashed line are overpredicted by the emulator. Points below the dashed line are under-predicted by the emulator.

Compared to streamflow time series, the distribution of model fidelities for the RFR algorithm-based emulator of groundwater depth variability exhibited better overall fidelity (**Figure 3A**). **Figures 3B and 3C** show the groundwater variability (as standard deviation over the 180-h simulation period) for the original model and the emulator, respectively. **Figure 3D** shows the emulator model error. In **Figures 3B ~ 3D**, the position of the urban patch is shown with a black outline. We observed that the largest positive (purple) and negative (orange) emulator errors tended to be exhibited in the midslope area of the domain (the highest elevation of the tilted slab domain is at the upper right corner, while the lower elevation is at the lower left corner) (**Figure 3D**). Emulator fidelity tended to be lower in scenarios where the uban patch spanned across the diagonal midslope portion of the domain, reflecting that dynamics in these areas might be more difficult to capture with an emulator. In addition, scenarios in which the shape of the urban patch were either the most elongated or the most square tended to have larger errors.

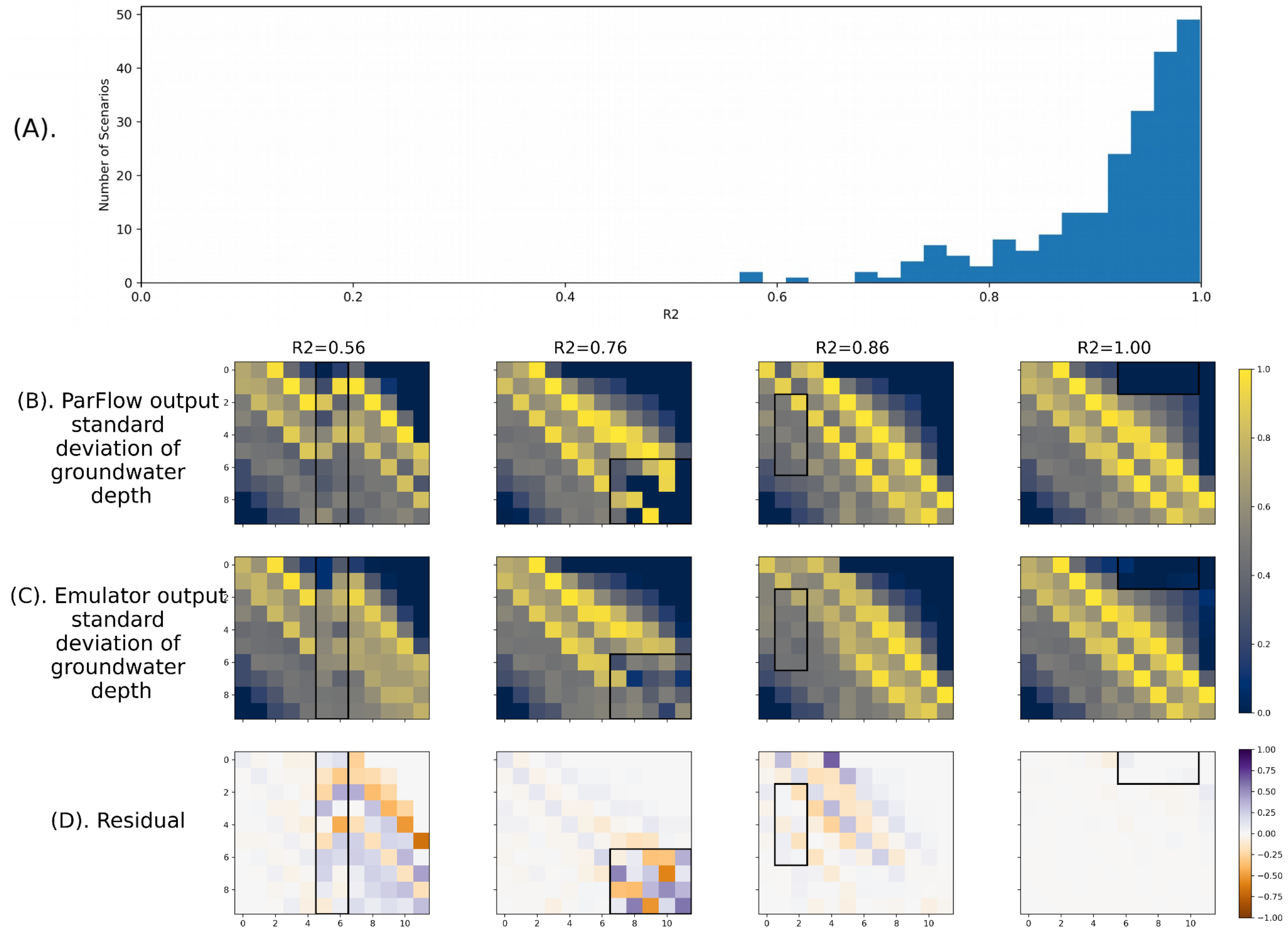


Figure 3A~C [color]. **3A)** Distribution of R^2 achieved by the RF-based emulator on 222 test set spatial scenarios. **3B)** Original model output for groundwater variability. **3C)** RF-based emulator prediction for GW variability. **3D)** Residual of the emulator compared to the original model.

4.2 Model robustness to training size

We assess the potential of the models in reducing the requirement for the number of simulated scenarios by tuning and testing models using various train/test split sizes and different sampling methods. This section analyzes the model robustness to the training sizes and in the following section, we analyze whether models can be improved by sampling. Model performance, which is evaluated using the percent of test scenarios for which emulators exhibited high fidelity to the original model (NSE > 0.7 for streamflow and $R^2 > 0.7$ for groundwater variability), is reported. (**Table 1**).

Table 1 Proportion of scenarios predicted with high fidelity with various training sizes

	Model performance on test set scenarios			
Training size	**10%**	**40%**	**50%**	**80%**
Temporal Streamflow Output				
KNN	0.50	0.73	0.74	0.82
Gaussian	0.29	0.70	0.75	0.88
RF	0.60	0.85	0.90	0.92
DNN	0.54	0.12	0.74	0.11
Spatial Groundwater Variability Output				
KNN	0.81	0.92	0.96	0.98
Gaussian	0.82	0.94	0.96	0.94
RF	0.82	0.96	0.98	0.97
DNN	0.61	0.55	0.74	0.88

For the streamflow output, RFR consistently produced the highest percent of high fidelity scenarios. KNN outperformed the GPR algorithm when the training size is small (i.e., 10%), while GPR had higher prediction accuracy when the training size is large (i.e., 80%). Even though many applications (Greenspan, van Ginneken, and Summers 2016; Marçais and Dreuzy 2017; Najafabadi et al. 2015) show that DNN can achieve better performance than other models, it requires complex setup and more computation resources to ensure a decent prediction accuracy. Since this study uses the simple implementation of DNN in Scikit-learn, it is within our expectation that the predictive power is very limited. In addition, we limited the model hyperparameter tuning processes to 10 hours, which affected DNN, since other algorithms' tuning processes were completed in well under 10 hours. As a result, only a few combinations of hyperparameters were tried for DNN due to extremely long execution time of several hours for one trial. This resulted in the percent of scenarios with high prediction fidelity from DNN to vary dramatically with the training sizes, indicating that the DNN models were not sufficiently tuned due to the expensive tuning process.

For the groundwater variability output, when the training sizes were small, RFR outperformed all other models while KNN showed a better performance with larger training size. RFR is also more generalizable according to a comparison of performance on test scenarios and training scenarios. Similar to the prediction for streamflow, DNN consistently made predictions with low accuracy and shows instability.

Overall, RFR showed higher fidelity to the original model with small training sizes compared to other models. Specifically, when the training set is only 10% of the whole data set, RFR ensured that over 60% and 80% of the scenarios are predicted with high fidelity for both the streamflow prediction and groundwater variability.

4.3 Model improvement with stratified sampling methods

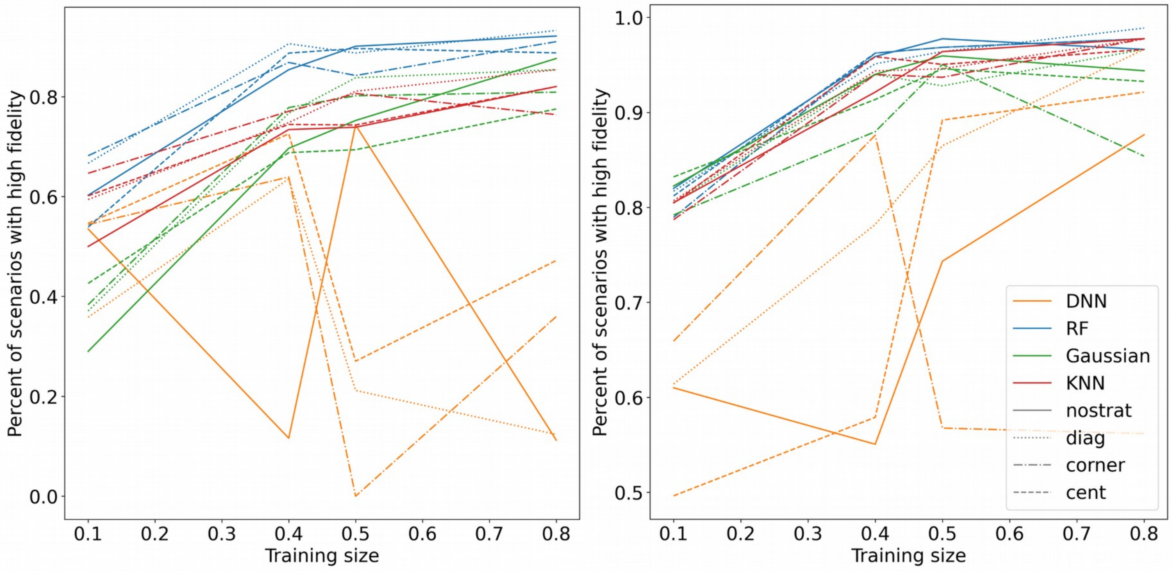
We hypothesized that using stratification on scenarios' input land use configurations when splitting the data into testing and training sets could help provide more information to the emulator training process without requiring an increased number of training scenarios. Therefore, we tested the models with various stratification methods using a 50% training size. However, as shown in **Table 2**, we found mixed evidence. For the temporal streamflow output, the prediction from KNN and GPR were improved while RFR and DNN did not benefit from stratified sampling. In contrast, we did not observe any improvement in the model performance for RFR, GPR, and KNN with stratification while stratifying on the distance of the urban patch's centroid to the lowest elevation point in the domain and on the size of the urban patch can improve the performance of DNN.

Table 2 Percent of test set scenarios where emulator achieved > 0.7 for NSE or spatial R^2 when stratifying samples on scenario characteristics in test-train splits

	No stratification	**Stratify on distance of urban patch centroid to domain's lowest elevation**	**Stratify on distance of urban patch corner to domain's lowest elevation**	**Stratify on length of diagonal of the urban patch**
Temporal Streamflow Output				
KNN	0.74	0.74	0.81	0.81
Gaussian	0.75	0.69	0.80	0.84
RF	0.90	0.90	0.84	0.89
DNN	0.74	0.27	0.00	0.21
Spatial Groundwater Variability Output				
KNN	0.96	0.95	0.94	0.95
Gaussian	0.96	0.95	0.95	0.93
RF	0.98	0.97	0.97	0.96
DNN	0.74	0.89	0.57	0.86

4.4 Interactive effect of the training size and sampling method on model performance

Lastly, we also evaluated how the training sizes and stratified sampling may work together to influence the four algorithms' performance. We conducted more experiments combining the effect of training sizes (i.e., 10%, 40%, 50%, and 80%) with stratification methods (i.e., stratifying on distance of urban patch centroid to domain's lowest elevation [cent], stratifying on distance of urban patch corner to domain's lowest elevation [corner], and stratifying on length of diagonal of the urban patch [diag]). The percent of test scenarios with high fidelity in each model with the combinations of training size and sampling method is visualized in **Figure 4**.



(a). Streamflow

(b). Groundwater variability

Figure 4. Percent of test scenarios predicted with high fidelity of all models

The results suggest that stratified sampling improves the model performance for predicting streamflow when there is limited training data. In particular, when only 10% of the scenarios are used as training data, the RFR model achieves almost 70% of the test scenarios predicted with high fidelity, if combined with the sampling method "corner," whereas if the test-train split is not assigned based on stratified sampling, the RFR algorithm achieves only about 60% of the test scenarios predicted with high fidelity. Therefore, stratification can help reduce the requirement for the number of simulation experiments while preserving a decent prediction accuracy, for the RFR algorithm. The benefit associated with both stratified and non-stratified samples on model performance improvement decreases as training sizes increase. The prediction accuracy levels off once the training size reaches 50% for RFR.

For groundwater variability, stratification does not significantly reduce the requirements for training data needed for each of the algorithms. This is probably due to an already high model prediction accuracy compared to streamflow. Similar to streamflow, about half of the simulation experiments are needed to reproduce the results with high fidelity.

5 Discussion

While we originally hypothesized that DNN would be able to capture many of the nonlinear responses caused by surface and groundwater interactions enabled through high-resolution coupled surface-subsurface hydrological model ParFlow.CLM, the results indicated that DNN did not result in the highest rates of high fidelity emulation. Instead, for both spatial and temporal outputs, the RFR algorithm produced the highest fidelity emulators with the highest levels of stability across random training samples. Upon inspection of the scenarios that were predicted with high fidelity and low fidelity, we found several factors that were correlated with low fidelity predictions: (1) scenarios where the "urban patch" intersected the midslope area of the hillslope domain, where saturation is highly variable over the simulation period; (2) scenarios where the shape of the urban patch is highly elongated, or in the case of predicting groundwater variability, very square. In the former factor, poor prediction performance is indeed due to the presence of event-specific nonlinearities in this zone. In the latter, poor prediction may be more attributed to the uniqueness of the geometries of the patch -- both the most elongated and most compact configurations represent the "extremes" of the overall population of scenarios.

Recently, there has been increasing attention given to DNNs, precisely for their potential to mimic high dimensional, nonlinear patterns. They have been particularly useful in object detection and computer vision applications (Eldan & Shamir, 2016; Greenspan et al., 2016; Najafabadi et al., 2015), and applied in hydrological modeling as well (Shen, 2018). However, in this application we found that DNNs were both difficult and costly to parameterize, and showed evidence that our training data sets (n = 444) were simply not large enough to support the algorithm's advantages. Hyperparameter selection is an important step in training ML algorithms and we utilized Neural Network Intelligence (NNI) to automate grid searches for optimal parameters for each algorithm we tested. The process of hyperparameter optimization was also more costly for DNN than for RFR. In the 10 hours of the model tuning time, only fewer than 10 trials (i.e., 10 combinations of the hyperparameters) were finished for DNN while more than 1000 trials can be tested for RFR. Testing of a range of training set sizes showed that DNN exhibited an increase in performance, which could indicate that this algorithm might continue to improve when provided more training data, as other studies have also noted (Cho et al., 2016; Ding et al., 2017). However, when combined with sample stratification, the tuned DNN hyperparameters exhibited instability in predicting scenarios in a test set. In the context of emulating for environmental decision-making contexts that motivated this study, running the original model enough times to provide data to ML emulators would simply not be a practical investment in most cases.

As others have noted, models must be evaluated differently in decision-making contexts than in contexts of pure scientific inquiry (Hamilton et al., 2019; White et al., 2010). We suggest that model emulators used in decision-making contexts must also be assessed in aspects additional to fidelity to the original model. Specifically, (1) does the ML emulator itself necessitate high levels of specialized data science expertise in order to ensure appropriate hyperparameter selection and stability?; and (2) does the emulator require so much training data to be generated from the original model so as to create its own barriers to implementation?

In the case of this study, the ML algorithm that produced the highest fidelity emulation neither required as much training data and nor was as complicated to tune than the more complex ML algorithm DNN. In more complex applications of high-resolution, spatially distributed hydrological models such as: complex geometries of land cover patches, more varied topographies and subsurface conditions however, this might not be the case. More research in emulation using ML algorithms is needed to better anticipate what algorithms are needed to emulate which levels of process complexities, as well as approximately how much training data might be required to ensure high fidelity and stability of the emulator.

6 Conclusion

In this study, we examined the processes of training four machine learning algorithm-based emulators of the high-resolution surface-subsurface hydrological model ParFlow.CLM. We were particularly interested in defining the emulators to take spatial scenarios of land use configurations (e.g. urban vs forested land covers) as inputs to support spatial environmental decision-making. Spatial inputs have not previously been a major emphasis of emulators, which have tended to be applied to rapidly mimic model outputs given varying input values of parameters. We found that the best fidelity to the original model was achieved by the random forest regressor, and that an algorithm hypothesized to be able to capture high dimensional patterns and nonlinearity in the system, a deep multilayer perceptron, did not exhibit high fidelity or stability given random samples of training data. Examination of the results indicated that this was probably due to a combination of the relatively simple application of the ParFlow.CLM model, not enough training data generated to support the DNN algorithm, and complexity of tuning hyperparameters. Lastly, we suggest that when applied for spatial decision support, choice of emulators need to consider more than the fidelity of reproducing the "response surface," and also consider necessary costs of data science expertise and generating sufficient training data.

Computer Code Availability

ParFlow.CLM (3.7.0)

- Developer: Reed Maxwell
- Year first available: 2001
- Software Required: None
- Program languages: C, Tcl, Python, Fortran
- Program size: 100.629 MB
- Access: Downloadable from: https://github.com/parflow/parflow

Code for pre/post processing of ParFlow inputs:

- Developer: Theodore Lim

- Year first available 2020
- Software Required: Python
- Program languages: Python
- Program size: 49 MB
- Access: Downloadable from: https://github.com/theochli/parflow_py_utils

Acknowledgements

This work was funded by the Institute for Culture, Society and Environment at the Virginia Polytechnic Institute and State University.

Appendix

Table 1. Search Space of the Model Hyperparameter Tuning

Model	**Hyperparameter**	**Search Space**
KNN	n_neighbors	[1,2,3,...,30]
	weights	[‘uniform’, ‘distance’]
	p	[1,2]
	metric	["euclidean", "manhattan", "chebyshev"]
Gaussian	constant_value	[0,1,2,3,...10]
	lower_bd	[1e-5, 1e2]
	upper_bd	[1e2, 1e5]
	length_scale	[1,2,3,...,20]
	length_scale_up	[1e3, 1e7]
	length_scale_low	[1e-3, 1e3]
	n_restarts_optimizer	[0,1,2,3...10]
	alpha	[1e-5, 1]
DNN	activation	["identity", "logistic", "tanh", "relu"]
	solver	["lbfgs", "sgd", "adam"]
	alpha	[1e-8, 1]
	batch_size	[10,11,12,…,800]
	learning_rate	["constant", "invscaling", "adaptive"]
	learning_rate_init	[1e-8, 1]
	power_t	[1e-8, 1]
	max_iter	1000
	shuffle	[0,1]
	tol	[1e-8, 1]
	momentum	[1e-8, 1]
	nesterovs_momentum	[0,1]
	validation_fraction	[0,0.95,0.05]
	beta_1	[1e-8,1]
	beta_2	[1e-8,1]
	epsilon	[1e-8,1]
	n_iter_no_change	[1,2,3,…,100]
RF	n_estimators	[1,2,…,200]
	max_depth	[5,6,7,…,60]

	min_samples_split	[2,3,4,5,6]
	min_samples_leaf	[1,2,3,4,5,6]

References

Alberti, M., & Booth, D. B. (2007). The impact of urban patterns on aquatic ecosystems: An empirical analysis in Puget lowland sub-basins. *Landscape and Urban Planning*, *80*(4), 345–361. https://doi.org/10.1016/j.landurbplan.2006.08.001

Arciniegas, G., Janssen, R., & Rietveld, P. (2013). Effectiveness of collaborative map-based decision support tools: Results of an experiment. *Environmental Modelling & Software*, 39, 159–175. https://doi.org/10.1016/j.envsoft.2012.02.021

Arnold, C. L., & Gibbons, C. J. (1996). Impervious Surface Coverage: The Emergence of a Key Environmental Indicator. *Journal of the American Planning Association*, *62*(2), 243–258. https://doi.org/10.1080/01944369608975688

Babaei, M., & Pan, I. (2016). Performance comparison of several response surface surrogate models and ensemble methods for water injection optimization under uncertainty. *Computers & Geosciences*, 91, 19–32. https://doi.org/10.1016/j.cageo.2016.02.022

Barnes, M. L., Welty, C., & Miller, A. J. (2018). Impacts of Development Pattern on Urban Groundwater Flow Regime. *Water Resources Research*, 54(8), 5198–5212. https://doi.org/10.1029/2017WR022146

Berke, P. R., Macdonald, J., White, N., Holmes, M., Line, D., Oury, K., & Ryznar, R. (2003). Greening Development to Protect Watersheds: Does New Urbanism Make a Difference? *Journal of the American Planning Association*, *69*(4), 397–413. https://doi.org/10.1080/01944360308976327

Beven, K. (1993). Prophecy, reality and uncertainty in distributed hydrological modelling. *Advances in Water Resources*, *16*(1), 41–51. https://doi.org/10.1016/0309-1708(93)90028-E

Bhaskar, A., Welty, C., Maxwell, R. M., & Miller, A. J. (2015). Untangling the effects of urban development on subsurface storage in Baltimore. *Water Resources Research*, *51*(2), 1158–1181. https://doi.org/10.1002/2014WR016039

Booth, D. B., & Jackson, C. R. (1997). Urbanization of Aquatic Systems: Degradation Thresholds, Stormwater Detection, and the Limits of Mitigation1. *JAWRA Journal of the American Water Resources Association*, *33*(5), 1077–1090. https://doi.org/10.1111/j.1752-1688.1997.tb04126.x

Boyd, M. J., Bufill, M. C., & Knee, R. M. (1994). Predicting pervious and impervious storm runoff from urban drainage basins. *Hydrological Sciences Journal*, *39*(4), 321–332. https://doi.org/10.1080/02626669409492753

Carnevale, C., Finzi, G., Guariso, G., Pisoni, E., & Volta, M. (2012). Surrogate models to compute optimal air quality planning policies at a regional scale. *Environmental Modelling & Software*, *34*, 44–50. https://doi.org/10.1016/j.envsoft.2011.04.007

Cho, J., Lee, K., Shin, E., Choy, G., & Do, S. (2016). How much data is needed to train a medical image deep learning system to achieve necessary high accuracy? *ArXiv:1511.06348 [Cs]*. http://arxiv.org/abs/1511.06348

Choat, B. E., & Bhaskar, A. S. (2020). Spatial Arrangement of Stormwater Infiltration Affects Subsurface Storage and Baseflow. *Journal of Hydrologic Engineering*, *25*(11), 04020048. https://doi.org/10.1061/(ASCE)HE.1943-5584.0002005

Chwif, L., Barretto, M. R. P., & Paul, R. J. (2000). On simulation model complexity. *2000 Winter Simulation Conference Proceedings (Cat. No.00CH37165)*, *1*, 449–455 vol.1. https://doi.org/10.1109/WSC.2000.899751

Clement, T. P. (2011). Complexities in Hindcasting Models—When Should We Say Enough Is

Enough? *Groundwater*, *49*(5), 620–629. https://doi.org/10.1111/j.1745-6584.2010.00765.x

Collick, A. S., Fuka, D. R., Kleinman, P. J. A., Buda, A. R., Weld, J. L., White, M. J., Veith, T. L., Bryant, R. B., Bolster, C. H., & Easton, Z. M. (2015). Predicting phosphorus dynamics in complex terrains using a variable source area hydrology model. *Hydrological Processes*, *29*(4), 588–601. https://doi.org/10.1002/hyp.10178

Cox, G. M., Gibbons, J. M., Wood, A. T. A., Craigon, J., Ramsden, S. J., & Crout, N. M. J. (2006). Towards the systematic simplification of mechanistic models. *Ecological Modelling*, *198*(1), 240–246. https://doi.org/10.1016/j.ecolmodel.2006.04.016

Crompton, O., Sytsma, A., & Thompson, S. (2019). Emulation of the Saint Venant Equations Enables Rapid and Accurate Predictions of Infiltration and Overland Flow Velocity on Spatially Heterogeneous Surfaces. *Water Resources Research*, *55*(8), 7108–7129. https://doi.org/10.1029/2019WR025146

De la Rosa, D., Mayol, F., Diaz-Pereira, E., Fernandez, M., & de la Rosa, D. (2004). A land evaluation decision support system (MicroLEIS DSS) for agricultural soil protection. *Environmental Modelling & Software*, *19*(10), 929–942. https://doi.org/10.1016/j.envsoft.2003.10.006

Ding, J., Li, X., & Gudivada, V. N. (2017). Augmentation and evaluation of training data for deep learning. *2017 IEEE International Conference on Big Data (Big Data)*, 2603–2611. https://doi.org/10.1109/BigData.2017.8258220

Dunne, T., Moore, T. R., & Taylor, C. H. (1975). Recognition and Prediction of Runoff-Producing Zones in Humid Regions. *Hydrological Sciences - Bulletin*, *20*(3), 305–327.

Easton, Z. M., Gérard-Marchant, P., Walter, M. T., Petrovic, A. M., & Steenhuis, T. S. (2007). Hydrologic assessment of an urban variable source watershed in the northeast United States. *Water Resources Research*, *43*(3), W03413. https://doi.org/10.1029/2006WR005076

Ebrahimian, A., Wilson, B. N., & Gulliver, J. S. (2016). Improved methods to estimate the effective impervious area in urban catchments using rainfall-runoff data. *Journal of Hydrology*, *536*, 109–118. https://doi.org/10.1016/j.jhydrol.2016.02.023

Eldan, R., & Shamir, O. (2016). *The Power of Depth for Feedforward Neural Networks*. *49*, 1–34.

Endreny, T., & Collins, V. (2009). Implications of bioretention basin spatial arrangements on stormwater recharge and groundwater mounding. *Ecological Engineering*, *35*(5), 670–677. https://doi.org/10.1016/j.ecoleng.2008.10.017

Epps, T. H., & Hathaway, J. M. (2018). Establishing a Framework for the Spatial Identification of Effective Impervious Areas in Gauged Basins: Review and Case Study. *Journal of Sustainable Water in the Built Environment*, *4*(2), 05018001. https://doi.org/10.1061/JSWBAY.0000853

Epps, T. H., & Hathaway, J. M. (2019). Using spatially-identified effective impervious area to target green infrastructure retrofits: A modeling study in Knoxville, TN. *Journal of Hydrology*, *575*, 442–453. https://doi.org/10.1016/j.jhydrol.2019.05.062

Forrester, A., Sobester, A., & Keane, A. (2008). *Engineering Design via Surrogate Modelling: A Practical Guide*. John Wiley & Sons.

Funtowicz, S. O., & Ravetz, J. R. (1993). The Emergence of Post-Normal Science. In R. Von Schomberg (Ed.), *Science, Politics and Morality*. Springer Netherlands. https://doi.org/10.1007/978-94-015-8143-1

Grayson, R. B., Moore, I. D., & McMahon, T. A. (1992). Physically based hydrologic modeling:

2. Is the concept realistic? *Water Resources Research*, *28*(10), 2659–2666. https://doi.org/10.1029/92WR01259

Greenspan, H., van Ginneken, B., & Summers, R. M. (2016). Guest Editorial Deep Learning in Medical Imaging: Overview and Future Promise of an Exciting New Technique. *IEEE Transactions on Medical Imaging*, *35*(5), 1153–1159. https://doi.org/10.1109/TMI.2016.2553401

Hamilton, S. H., Fu, B., Guillaume, J. H. A., Badham, J., Elsawah, S., Gober, P., Hunt, R. J., Iwanaga, T., Jakeman, A. J., Ames, D. P., Curtis, A., Hill, M. C., Pierce, S. A., & Zare, F. (2019). A framework for characterising and evaluating the effectiveness of environmental modelling. *Environmental Modelling & Software*, *118*, 83–98. https://doi.org/10.1016/j.envsoft.2019.04.008

Hammer, T. R. (1972). Stream channel enlargement due to urbanization. *Water Resources Research*, *8*(6), 1530–1540. https://doi.org/10.1029/WR008i006p01530

Hong, E.-M., Pachepsky, Y. A., Whelan, G., & Nicholson, T. (2017). Simpler models in environmental studies and predictions. *Critical Reviews in Environmental Science and Technology*, *47*(18), 1669–1712. https://doi.org/10.1080/10643389.2017.1393264

Inman, D., Blind, M., Ribarova, I., Krause, A., Roosenschoon, O., Kassahun, A., Scholten, H., Arampatzis, G., Abrami, G., McIntosh, B., & Jeffrey, P. (2011). Perceived effectiveness of environmental decision support systems in participatory planning: Evidence from small groups of end-users. *Environmental Modelling & Software*, *26*(3), 302–309. https://doi.org/10.1016/j.envsoft.2010.08.005

Jones, D., Jones, N., Greer, J., & Nelson, J. (2015). A cloud-based MODFLOW service for aquifer management decision support. *Computers & Geosciences*, 78, 81–87. https://doi.org/10.1016/j.cageo.2015.02.014

Klosterman, R. E. (1999). The What If? Collaborative Planning Support System. *Environment and Planning B: Planning and Design*, *26*(3), 393–408. https://doi.org/10.1068/b260393

Korfmacher, K. S. (2001). The Politics of Participation in Watershed Modeling. *Environmental Management*, *27*(2), 161–176. https://doi.org/10.1007/s002670010141

Laniak, G. F., Olchin, G., Goodall, J., Voinov, A., Hill, M., Glynn, P., Whelan, G., Geller, G., Quinn, N., Blind, M., Peckham, S., Reaney, S., Gaber, N., Kennedy, R., & Hughes, A. (2013). Integrated environmental modeling: A vision and roadmap for the future. *Environmental Modelling & Software*, *39*, 3–23. https://doi.org/10.1016/j.envsoft.2012.09.006

Layzer, J. A. (2011). Ecosystem-Based Management in the Chesapeake Bay. In *The Environmental Case: Translating Values Into Policy*. CQ Press.

Lee, D. B. (1973). Requiem for large-scale models. *Journal of the American Institute of Planners*, *39*(3).

Leonard, L., Miles, B., Heidari, B., Lin, L., Castronova, A. M., Minsker, B., Lee, J., Scaife, C., & Band, L. E. (2019). Development of a participatory Green Infrastructure design, visualization and evaluation system in a cloud supported jupyter notebook computing environment. *Environmental Modelling & Software*, *111*, 121–133. https://doi.org/10.1016/j.envsoft.2018.10.003

Lerner, D. N. (2002). Identifying and quantifying urban recharge: A review. *Hydrogeology Journal*, *10*(1), 143–152. https://doi.org/10.1007/s10040-001-0177-1

Lim, T. (2016). Predictors of urban variable source area: A cross-section analysis of urbanized catchments in the united states. *Hydrological Processes*, 4799–4814. https://doi.org/10.1002/hyp.10943

Lim, T. C., & Welty, C. (2017). Effects of spatial configuration of imperviousness and green infrastructure networks on hydrologic response in a residential sewershed. *Water Resources Research*, *53*(9), 8084–8104. https://doi.org/10.1002/2017WR020631

Little, J. C., Hester, E. T., Elsawah, S., Filz, G. M., Sandu, A., Carey, C. C., Iwanaga, T., & Jakeman, A. J. (2019). A tiered, system-of-systems modeling framework for resolving complex socio-environmental policy issues. *Environmental Modelling & Software*, *112*, 82–94. https://doi.org/10.1016/j.envsoft.2018.11.011

Liu, Y., Gupta, H., Springer, E., & Wagener, T. (2008). Linking science with environmental decision making: Experiences from an integrated modeling approach to supporting sustainable water resources management. *Environmental Modelling & Software*, *23*(7), 846–858. https://doi.org/10.1016/j.envsoft.2007.10.007

Mahmoud, M., Liu, Y., Hartmann, H., Stewart, S., Wagener, T., Semmens, D., Stewart, R., Gupta, H., Dominguez, D., Dominguez, F., Hulse, D., Letcher, R., Rashleigh, B., Smith, C., Street, R., Ticehurst, J., Twery, M., van Delden, H., Waldick, R., … Winter, L. (2009). A formal framework for scenario development in support of environmental decision-making. *Environmental Modelling & Software*, *24*(7), 798–808. https://doi.org/10.1016/j.envsoft.2008.11.010

Maxwell, R. M., & Miller, N. L. (2005). Development of a Coupled Land Surface and Groundwater Model. *Journal of Hydrometeorology*, *6*(3), 233–247. https://doi.org/10.1175/JHM422.1

Maxwell, R. M., Putti, M., Meyerhoff, S., Delfs, J.-O., Ferguson, I. M., Ivanov, V., Kim, J., Kolditz, O., Kollet, S. J., Kumar, M., Lopez, S., Niu, J., Paniconi, C., Park, Y.-J., Phanikumar, M. S., Shen, C., Sudicky, E. A., & Sulis, M. (2014). Surface-subsurface model intercomparison: A first set of benchmark results to diagnose integrated hydrology and feedbacks. *Water Resources Research*, *50*(2), 1531–1549. https://doi.org/10.1002/2013WR013725

McDonnell, J. J., Sivapalan, M., Vaché, K., Dunn, S., Grant, G., Haggerty, R., Hinz, C., Hooper, R., Kirchner, J., Roderick, M. L., Selker, J., & Weiler, M. (2007). Moving beyond heterogeneity and process complexity: A new vision for watershed hydrology. *Water Resources Research*, *43*(7). https://doi.org/10.1029/2006WR005467

Microsoft Research (MSR). (2021). *Neural Network Intelligence—An open source AutoML toolkit for neural architecture search, model compression and hyper-parameter tuning (NNI v2.0)*. https://nni.readthedocs.io/en/stable/

Miles, B., & Band, L. E. (2015). Green infrastructure stormwater management at the watershed scale: Urban variable source area and watershed capacitance. *Hydrological Processes*, *29*(9), 2268–2274. https://doi.org/10.1002/hyp.10448

Mo, S., Zabaras, N., Shi, X., & Wu, J. (2019). Deep Autoregressive Neural Networks for High-Dimensional Inverse Problems in Groundwater Contaminant Source Identification. *Water Resources Research*, *55*(5), 3856–3881. https://doi.org/10.1029/2018WR024638

Mo, S., Shi, X., Lu, D., Ye, M., & Wu, J. (2019). An adaptive Kriging surrogate method for efficient uncertainty quantification with an application to geological carbon sequestration modeling. *Computers & Geosciences*, *125*, 69–77. https://doi.org/10.1016/j.cageo.2019.01.012

Moreno-Rodenas, A. M., Bellos, V., Langeveld, J. G., & Clemens, F. H. L. R. (2018). A dynamic emulator for physically based flow simulators under varying rainfall and parametric conditions. *Water Research*, *142*, 512–527. https://doi.org/10.1016/j.watres.2018.06.011

Moriasi, D. N., Arnold, J. G., Liew, M. W. V., Bingner, R. L., Harmel, R. D., & Veith, T. L. (2007).

MODEL EVALUATION GUIDELINES FOR SYSTEMATIC QUANTIFICATION OF ACCURACY IN WATERSHED SIMULATIONS. *TRANSACTIONS OF THE ASABE*, *50*, 16.

Najafabadi, M. M., Villanustre, F., Khoshgoftaar, T. M., Seliya, N., Wald, R., & Muharemagic, E. (2015). Deep learning applications and challenges in big data analytics. *Journal of Big Data*, *2*(1), 1. https://doi.org/10.1186/s40537-014-0007-7

Oleson, K. W., Lawrence, D. M., Bonan, G. B., Kluzek, E., Thornton, P. E., Dai, A., Decker, M., Dickinson, R., Feddema, J., Heald, C. L., Hoffman, F., Lamarque, J.-F., Mahowald, N., Niu, G.-Y., Qian, T., Randerson, J., Running, S., Sakaguchi, K., Slater, A., … Zeng, X. (2010). *Technical Description of version 4.0 of the Community Land Model (CLM)* (ISSN Print Edition 2153-2397). National Center for Atmospheric Research. http://www.cesm.ucar.edu/models/ccsm4.0/clm/CLM4_Tech_Note.pdf

Qiu, Z. (2009). Assessing Critical Source Areas in Watersheds for Conservation Buffer Planning and Riparian Restoration. *Environmental Management*, *44*(5), 968–980. https://doi.org/10.1007/s00267-009-9380-y

Ratto, M., Castelletti, A., & Pagano, A. (2012). Emulation techniques for the reduction and sensitivity analysis of complex environmental models. *Environmental Modelling & Software*, *34*, 1–4. https://doi.org/10.1016/j.envsoft.2011.11.003

Razavi, S., Tolson, B. A., & Burn, D. H. (2012). Review of surrogate modeling in water resources: REVIEW. *Water Resources Research*, *48*(7). https://doi.org/10.1029/2011WR011527

Schueler, T. (1994). The Importance of Imperviousness. *Watershed Protection Techniques*, *1*(3), 100–111.

Schueler, T., Fraley-McNeal, L., & Cappiella, K. (2009). Is Impervious Cover Still Important? Review of Recent Research. *Journal of Hydrologic Engineering*, *14*(4), 309–315. https://doi.org/10.1061/(ASCE)1084-0699(2009)14:4(309)

Schwartz, F. W., Liu, G., Aggarwal, P., & Schwartz, C. M. (2017). Naïve Simplicity: The Overlooked Piece of the Complexity-Simplicity Paradigm. *Groundwater*, *55*(5), 703–711. https://doi.org/10.1111/gwat.12570

Shen, C. (2018). A Transdisciplinary Review of Deep Learning Research and Its Relevance for Water Resources Scientists. *Water Resources Research*, *54*(11), 8558–8593. https://doi.org/10.1029/2018WR022643

Smith, B. K., & Smith, J. A. (2015). The Flashiest Watersheds in the Contiguous United States. *Journal of Hydrometeorology*, *16*(6), 2365–2381. https://doi.org/10.1175/JHM-D-14-0217.1

Steedman, R. J. (1988). Modification and Assessment of an Index of Biotic Integrity to Quantify Stream Quality in Southern Ontario. *Canadian Journal of Fisheries and Aquatic Sciences*, *45*(3), 492–501. https://doi.org/10.1139/f88-059

Tague, C., & Pohl-Costello, M. (2008). The Potential Utility of Physically Based Hydrologic Modeling in Ungauged Urban Streams. *Annals of the Association of American Geographers*, *98*(4), 818–833. https://doi.org/10.1080/00045600802099055

Tian, L., Wilkinson, R., Yang, Z., Power, H., Fagerlund, F., & Niemi, A. (2017). Gaussian process emulators for quantifying uncertainty in CO2 spreading predictions in heterogeneous media. *Computers & Geosciences*, 105, 113–119. https://doi.org/10.1016/j.cageo.2017.04.006

Urban, N. M., & Fricker, T. E. (2010). A comparison of Latin hypercube and grid ensemble designs for the multivariate emulation of an Earth system model. *Computers &*

Geosciences, 36(6), 746–755. https://doi.org/10.1016/j.cageo.2009.11.004

Voinov, A., & Bousquet, F. (2010). Modelling with stakeholders. *Environmental Modelling & Software*, *25*(11), 1268–1281. https://doi.org/10.1016/j.envsoft.2010.03.007

Walsh, C. J., Roy, A. H., Feminella, J. W., Groffman, P. M., & Morgan, R. P. (2005). The urban stream syndrome: Current knowledge and the search for a cure. *Journal of the North American Benthological Society*, *24*(3), 706–723.

Watson, L. T., Lohani, V. K., Kibler, D. F., Dymond, R. L., Ramakrishnan, N., & Shaffer, C. A. (2002). Integrated Computing Environments for Watershed Management. *Journal of Computing in Civil Engineering*, *16*(4), 259–268. https://doi.org/10.1061/(ASCE)0887-3801(2002)16:4(259)

White, D. D., Wutich, A., Larson, K. L., Gober, P., Lant, T., & Senneville, C. (2010). Credibility, salience, and legitimacy of boundary objects: Water managers' assessment of a simulation model in an immersive decision theater. *Science and Public Policy*, *37*(3), 219–232. https://doi.org/10.3152/030234210X497726

Wu, B., Zheng, Y., Wu, X., Tian, Y., Han, F., Liu, J., & Zheng, C. (2015). Optimizing water resources management in large river basins with integrated surface water-groundwater modeling: A surrogate-based approach. *Water Resources Research*, *51*(4), 2153–2173. https://doi.org/10.1002/2014WR016653

Wu, H., Bolte, J. P., Hulse, D., & Johnson, B. R. (2015). A scenario-based approach to integrating flow-ecology research with watershed development planning. *Landscape and Urban Planning*, *144*, 74–89. https://doi.org/10.1016/j.landurbplan.2015.08.012

Xiang, W.-N., & Clarke, K. C. (2003). The Use of Scenarios in Land-Use Planning. *Environment and Planning B: Planning and Design*, *30*(6), 885–909. https://doi.org/10.1068/b2945

Yang, J., Jakeman, A., Fang, G., & Chen, X. (2018). Uncertainty analysis of a semi-distributed hydrologic model based on a Gaussian Process emulator. *Environmental Modelling & Software*, *101*, 289–300. https://doi.org/10.1016/j.envsoft.2017.11.037

Zellner, M., & Campbell, S. D. (2015). Planning for deep-rooted problems: What can we learn from aligning complex systems and wicked problems? *Planning Theory & Practice*, *16*(4), 457–478. https://doi.org/10.1080/14649357.2015.1084360