WIREs Wiley Interdisciplinary Reviews

Machine Learning For Hydrologic Sciences: An Introductory Overview

Article Type:

OPINION | PRIMER | **OVERVIEW** (selected)
ADVANCED REVIEW | FOCUS ARTICLE | SOFTWARE FOCUS

Authors:

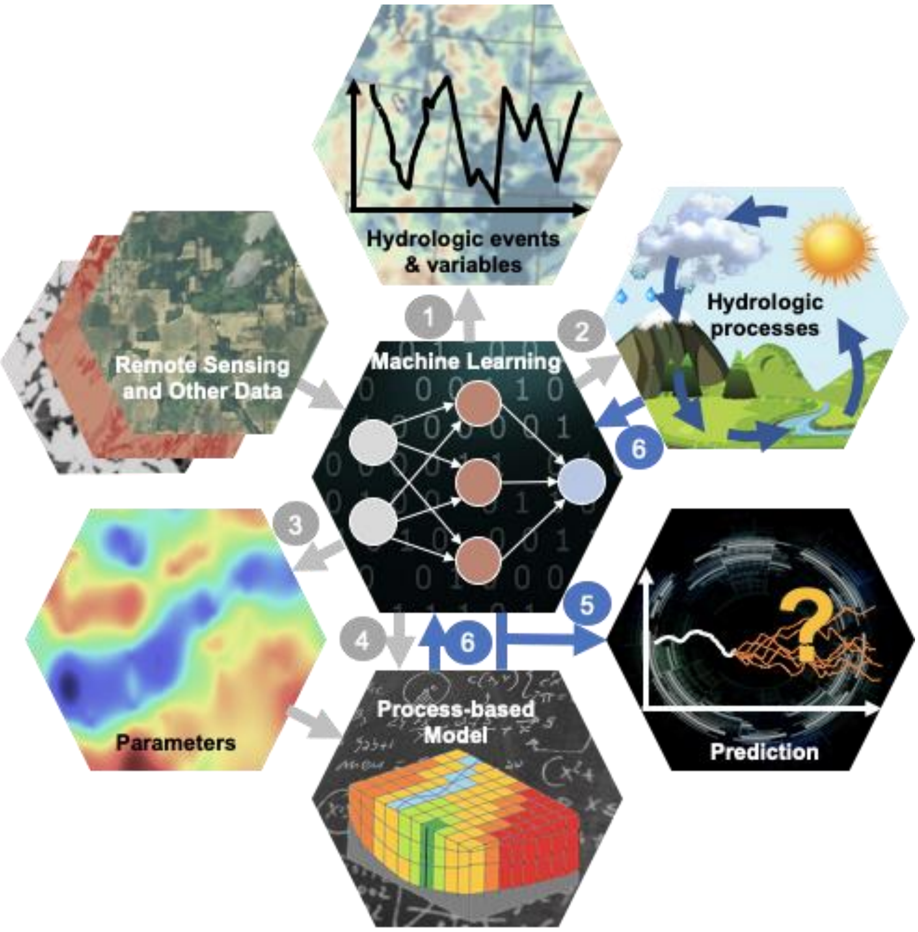
Tianfang Xu*
School of Sustainable Engineering and the Built Environment, Arizona State University, tianfang.xu@asu.edu. ORCID: 0000-0002-9565-9208

Feng Liang
Department of Statistics, University of Illinois at Urbana-Champaign, liangf@illinois.edu. ORCID: 0000-0002-4173-3003

Abstract

The hydrologic community has experienced a surge in interest in machine learning in recent years. This interest is primarily driven by rapidly growing hydrologic data repositories, as well as success of machine learning in various academic and commercial applications, now possible due to increasing accessibility to enabling hardware and software. This overview is intended for readers new to the field of machine learning. It provides a non-technical introduction, placed within a historical context, to commonly used machine learning algorithms and deep learning architectures. Applications in hydrologic sciences are summarized next, with a focus on recent studies. They include the detection of patterns and events such as land use change, approximation of hydrologic variables and processes such as rainfall-runoff modeling, and mining relationships among variables for identifying controlling factors. The use of machine learning is also discussed in the context of integrated with process-based modeling for parameterization, surrogate modeling, and bias correction. Finally, the article highlights challenges of extrapolating robustness, physical interpretability, and small sample size in hydrologic applications.

Graphical/Visual Abstract and Caption



Caption: Machine learning has been used in various hydrologic applications in stand-alone mode or integrated with process-based modeling. Arrows indicate information flow.

1. INTRODUCTION

Machine learning is the set of methods and algorithms that enable computers to automatically improve performance through experience. As such, they manifest the "data-driven" reasoning as opposed to "knowledge-driven" reasoning that underpins most physical science disciplines. Since the pioneering research that was conducted in the 1950s (Turing, 1950; Rosenblatt, 1958), the field of machine learning has seen dramatic progress. In the 1980s, backpropagation (Rumelhart et al., 1986) was found to be effective in training artificial neural networks (ANNs), which led to a surge in machine learning research centered around ANNs and their widespread applications in various disciplines, including hydrology (Buch et al., 1993; Kang et al., 1993; Hsu et al., 1995; Smith and Eli, 1995). Later, support vector machines (SVM, Vapnik, 1995) and other kernel methods (Liang et al., 2007; Hofmann et al., 2008) were discovered and became popular. In recent years, machine learning has become an interdisciplinary area intersecting with computer science, statistics, applied mathematics, and optimization.

Successful applications of conventional machine learning algorithms typically require a set of customized input features that best represent the raw data for the subsequent learning tasks. Deep learning, a class of machine learning algorithms based on ANNs of multiple layers (thus deep), is capable of automatically discovering appropriate representations from raw data (LeCun et al., 2015). While some deep learning architectures such as Recurrent Neural Network (RNN) were invented by the 1990s, widespread interest in deep learning research and applications flourished in the 2010s when low-cost computation and massive online data became increasingly available. Recent advances in machine learning, primarily in the field of deep learning, have brought breakthroughs in computer vision, speech recognition, and natural language processing and have achieved enormous successes in both scientific and commercial applications.

Inspired by the enormous success reported in the deep learning community and industry, researchers from various scientific disciplines are eager to apply machine learning techniques to problems from their own fields (Ching et al., 2018; Khan and Yairi, 2018; Radovic et al., 2018; Mater and Coote, 2019; Reichstein et al., 2019; Brunton et al., 2020; Sengupta et al., 2020). In the hydrologic sciences community, a growing interest in machine learning is largely driven by the availability of vast hydrologic data repositories (Shen, 2018; Shen et al., 2018). Advances in sensor technology, promotion of hydrologic observatories, and developments of cyberinfrastructure that enables easy sharing of data, have all ushered in an era of data deluge in the form of a plethora of *in situ* sensor measurements as well as remote sensing imagery. Existing knowledge about hydrological processes is, therefore, no longer adequate to represent the full range of variability observed in data (Hipsey et al., 2015; Kumar, 2015). In addition, due to the unprecedented volume and complexity of data, the knowledge-driven reasoning alone is not adequate to get the most out of available data. Machine learning, as well as the data-driven reasoning it enables, thus provides exciting opportunities for both the recovery of a full range of variability (thus bringing potentially improved prediction capability) as well as our capacity to discover new knowledge.

This paper aims to give a broad and non-technical overview of machine learning and its recent applications in hydrologic sciences. We begin this overview by introducing fundamental concepts and terminology. We then briefly describe several popular non-deep machine learning algorithms and deep learning architectures along with common practices of applying these methods. Next, we explore existing research, with a focus on recent studies that apply machine learning in hydrologic sciences. Finally, we conclude with challenges associated with applying machine learning for hydrologic problems and accompanying research opportunities.

2. MACHINE LEARNING BASICS

As a subset of artificial intelligence (AI), machine learning algorithms can automatically improve their performance with respect to some tasks through experience (Fig. 1; Mitchell, 1997). The experience here refers to examples or data points that are provided to the machine learning algorithm. An example consists of measurements of p input variables $\mathbf{x} = [x_1, \ldots, x_p]^T$; it may also contain a label or target, y, associated with x. *Unsupervised learning* aims to identify the underlying structure of the examples $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_\mathrm{n}\}$. On the other hand, *supervised learning* seeks to infer a function that maps inputs $\boldsymbol{x}$ to the label or target y. Supervised learning tasks can be further categorized into *classification* (when the labels take categorical values) and *regression* (when the labels take numerical values). For *supervised learning*, the performance refers to the discrepancy between the observed label or target and the one output by the learning algorithm. For *unsupervised learning*, since no label is available, the performance is often defined to be some objective function tied to the underlying algorithm. Another important consideration is how to represent the knowledge learned from experience. A machine learning algorithm makes assumptions about the functional form of the knowledge learned from experience, often referred to as the *hypothesis space*. Parametric machine learning algorithms make explicit assumptions regarding the format of the function, such as a linear or polynomial function of the input. In contrast, nonparametric alternatives tend to make less assumptions about the form of functions. For quick reference, Table 1 summarizes the above and other key terminology that will be discussed in this section.

Table 1. Definition of terms

Term	Explanation
Artificial intelligence	The study of intelligence demonstrated by a machine manifested by its capability to perceive the environment and take actions to achieve its goals and tasks through flexible adaptation (Kaplan and Haenlein, 2019).
Classification	A subtype of supervised learning where the targets are categories or labels.
Deep learning	A class of machine learning algorithms based on artificial neural networks (ANNs) and using hierarchical architectures to extract higher level features from input data via representation learning.
Feature engineering	The process of creating features from raw data that may be useful for subsequent learning task; typically implemented manually with domain expertise.
Generalization error/test error	The prediction capability of a trained machine learning model on independent *test* data unseen during training.
Hyperparameters/tuning parameters	Settings that can be tweaked to change the structure (e.g., number of layers in an ANN) and behavior (e.g., smoothness preference) of the learning algorithm.
Machine learning	A subset of AI (Fig. 1); learning methods and algorithms that enable computers to automatically improve performance through experience.
Overfitting	Overfitting occurs when a machine learning model has a high degree of freedom that cannot be fully justified by the training data. The opposite, underfitting, occurs when a model is too simple and thus inflexible in representing the range of variability of the training data.
Regression	A subtype of supervised learning where the targets are real numbers.
Regularization	A technique intended to reduce the generalization error, often by modifying the loss function to penalize deviation from certain preference (e.g., smoothness).
Representation learning	Techniques that automatically discover representation (or features) that are useful for subsequent learning tasks. Also known as feature learning.

Supervised learning	The computer is given examples consisting of inputs and their desired targets; the computer is *trained* on these examples to learn the input-to-target relationship.
Unsupervised learning	The computer is given inputs but no target variables; the goal is to find underlying patterns in the input data.

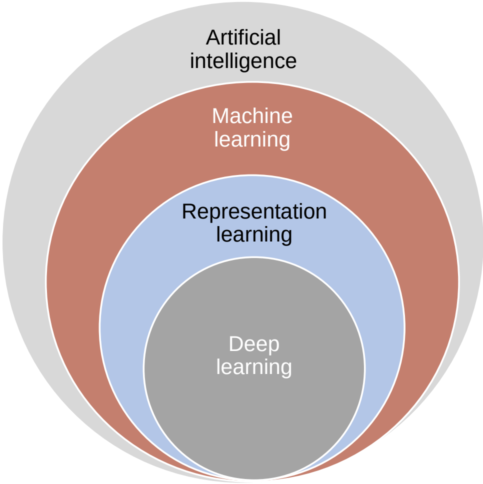


Figure 1. The nested concepts of artificial intelligence, machine learning, representation learning, and deep learning. Definitions of the four terms are listed in Table 1.

In the context of hydrology, unsupervised learning techniques can be used, for example, to cluster catchments into groups with distinct hydrologic regimes. Distinguishing different land cover types from multi-spectral satellite images can be formulated as a classification problem, where a classifier needs to learn the mapping from spectral bands and derived indices (inputs) to land cover classes (labels). A formulation of streamflow forecasting is a regression problem that learns a functional relationship between streamflow with some lead time (target) and inputs such as the past and forecasted meteorological conditions and past streamflow data. Given historical examples of the inputs and corresponding target, a machine learning model can be trained by minimizing mean squared error (performance metric). These problems can be approached using various machine learning algorithms that differ in the choices of hypothesis space, loss/objective function, and optimization method. Below we provide a brief, intuitive descriptions (along with references) of several conventional machine learning and deep learning algorithms that have been applied in hydrologic sciences. Readers are also referred to Shen et al. (2018) for a transdisciplinary review of deep learning and Tahmasebi et al. (2020) for a review of machine learning algorithms commonly used in geosciences focused on porous media problems. Readers who are interested in a more comprehensive, in-depth discussion of machine learning theory and algorithms may refer to Mitchell (1997), Hastie et al. (2009), and Goodfellow et al. (2016). Besides, Géron (2019) provides hands-on guide to machine learning and deep learning with working code.

2.1. Conventional machine learning algorithms

2.1.1. Clustering

Clustering, or cluster analysis, refers to a category of unsupervised learning methods that partitions data into groups with the goal of maximizing the similarity of data within the same group and minimizing the similarity of data among groups. There exist a variety of clustering methods and associated similarity measures, often based on the reciprocal of distance (Irani et al., 2016). A popular clustering algorithm is K-means, which takes a random initialization of the cluster assignment, and then iteratively minimizes the within-cluster point scatter until convergence (MacQueen, 1967; Hartigan and Wong, 1979). The within-cluster point scatter is defined as the sum of the distance (e.g., Euclidean) between every pair of data points assigned to the same cluster.

Over the past few decades, variants of K-means and other algorithms such as agglomerative hierarchical clustering and fuzzy clustering have been proposed and used in various applications (de Oliveira and Pedrycz, 2007; Jain, 2010; Murtagh and Legendre, 2014; Tennant et al., 2021). Although clustering is an unsupervised learning technique, it is sometimes used to learn data representation in the pre-processing step for a supervised learning task. For example, the cluster assignment can be used to produce new features on top of the raw input variables (Coates et al., 2011).

2.1.2. Lasso

Least Absolute Shrinkage and Selection Operator (Lasso) is a widely used regression method that adds an L_1 penalty term (the sum of absolute value of linear regression coefficients) to the ordinary least squares loss function in order to keep the regression coefficients small (Tibshirani, 1996). Because of the L_1 regularization, Lasso typically sets some of the regression coefficients to zero. The number of zero coefficients depends on the penalty hyperparameter, which is usually determined through cross validation. As such, the algorithm performs both feature selection and parameter estimation simultaneously, and has been widely used for high dimensional regression problems. In addition, Lasso can be used for classification when combined with logistic regression (Hosmer et al., 2013). Due to its good generalization performance, sparsity and interpretability, Lasso has been used in various applications (e.g., Anda et al., 2018; Bardsley et al., 2015; Vandal et al., 2019).

Table 2. Comparison of the representation of input variables by five supervised machine learning algorithms (Lasso, SVM, GPR, CART, and ANN).

Algorithm	Representation
Lasso	$\mathbf{x} = [x_1, \ldots, x_p]^T$, original inputs
SVM & GPR	$\phi(\mathbf{x})$, inputs projected to a higher dimensional feature space
CART	$\mathbf{1}\{\mathbf{x} \in R_i\}$, indicator function that equals 1 if $\mathbf{x}$ is in the leaf R_i and 0 otherwise.
ANN	$f_d\left(\ldots f_2\left(f_1(\mathbf{x})\right)\right)$, output of the last hidden layer

2.1.3. Support vector machine (SVM)

Support vector machine (SVM) is believed to be among the most robust prediction methods because it seeks to minimize an upper bound of the generalization error rather than the training error (Vapnik, 1995). In addition, the solution is globally optimal under conditions that can often be met, while other machine learning algorithms such as ANN may converge to local minima. The SVM algorithm maps the input variables to a higher dimensional feature space, $\phi(\mathbf{x})$ (Table 2). The map is usually implemented implicitly via a kernel function, also known as the kernel trick. The kernel function is analogous to the covariance function in Gaussian process (Section 2.1.4). For classification tasks, SVM identifies the optimal separating hyperplanes in the feature space while maximizing the margin between classes. Kernel trick enables SVM to classify data points that are not linearly separable in the original input space. For regression tasks, SVM minimizes an objective function composed of loss greater than a specified threshold and a L_2 regularization term. Ideally, the choice of kernel function should be made based on structure of the input data and their relation to the output. Lastly, it is worth noting that the model produced by SVM is represented sparsely as the linear combination of a subset of the training data ("support vectors") projected into the feature space.

2.1.4. Gaussian process regression

Gaussian process regression (GPR) is a Bayesian kernel regression method and has been shown to perform well in a variety of benchmark applications. A GP refers to a set of random variables, indexed in space and time, that have a joint multivariate Gaussian distribution. A GP is fully specified by a mean function and a covariance function that describes the covariance between each pair of the random variables (i.e., the quantity of interest at two separate locations/times). The two functions should reflect the prior knowledge of the general trend and level of smoothness of the target function, respectively. The use of covariance function is analogous to the kernel trick of SVM (Rasmussen and Williams, 2006) and implicitly maps the inputs to features $\phi(\mathbf{x})$ (Table 2). GP is also used by kriging methods in geostatistics, where the mean and covariance are typically specified as functions of spatial coordinates. In the context of machine learning, the independent variables of mean and covariance functions include explanatory variables, thus enabling GPR to approximate complex, nonlinear relationships between the target and inputs (features). Starting from the *a priori* (i.e., before seeing any data) mean and covariance, GPR uses the Bayes' Theorem to infer the posterior distribution of the target conditioned on the training data. Fig. 2a shows samples drawn from a GP with a mean that *a priori* follows a linear function of the input; in practical applications such prior knowledge should be incorporated when available. After training data is introduced, samples can be drawn from the posterior of the GP conditioned on training data (Fig. 2b). As such, GP regression is a probabilistic approach that explicitly derives the uncertainty associated with the predictions. As the test data moves away from the range of training data, the prediction given by GPR will converge to the prior mean with a wide prediction interval (uncertainty) (Fig. 2b). This is sometimes a preferred behavior when extrapolating with a function such as polynomial may lead to problematic results. Unlike the sparsity of SVM, exact GPR prediction at an unseen data point is a linear combination of all training data points, with the weights estimated based on the covariance function. Therefore, a disadvantage of GPR is that its computational cost with maintaining and operation of the covariance matrix can be prohibitive for large datasets. To overcome this difficulty and improve GPR scalability for big data, various approximation methods have been developed (Liu et al., 2020).

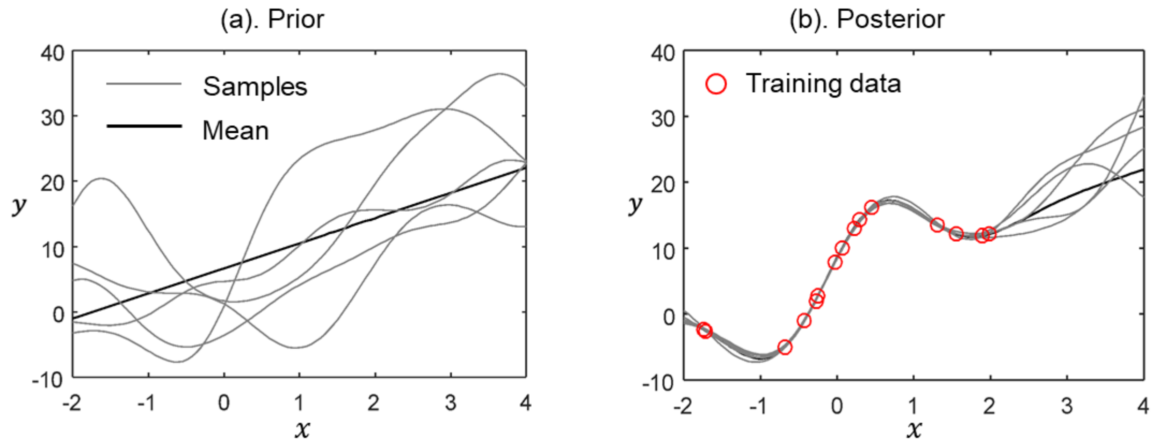


Figure 2. Schematic of Gaussian process regression (GPR) showing the (a) prior based on a linear mean function and a squared exponential covariance function, and (b) posterior conditioned on training data. Dark line shows the prior and posterior means, respectively, and grey lines are random samples drawn from the GP. Red open circles are training data points, and they "sculpt" the prior into the posterior.

2.1.5. Decision trees and forests

Decision trees are a conceptually simple nonparametric machine learning algorithm. Here we briefly describe the classification and regression trees (CARTs). A CART recursively partitions the feature space into rectangular regions using a sequence of binary splits. Each time, the CART chooses a splitting variable from all input variables and threshold to maximize the goodness-of-fit after this split. The process is repeated until a user-specified minimum number of data points is reached at the leaves, or terminal nodes. Each leaf represents a rectangle region in the input space, denoted as $R_i, i = 1, \ldots, N$ with N denoting the total number of leaves, and CART fits a constant value α_i to R_i. For an unseen data point $\mathbf{x}^*$, CART prediction is a linear combination of the values of each leaf, i.e. $\sum_{i=1}^{N} \alpha_i \mathbf{1}\{\mathbf{x}^* \in R_i\}$, , where $\mathbf{1}\{\mathbf{x} \in R_i\}$ is an indicator function equal to 1 if $\mathbf{x}^*$ falls within the i-th leaf and zero otherwise (Table 2). In its essence, a CART estimates a piecewise constant function. It is a common practice to prune the tree to a subtree to prevent overfitting. A major advantage of decision trees is their interpretability. One disadvantage of decision trees is their statistical instability even after pruning. In other words, small perturbation or noise in the training data may result in substantially different structure of the learned tree (Hastie et al., 2009).

To overcome the aforementioned disadvantage, forests that are based on multiple trees have been proposed. For example, the random forests (RF) are an ensemble learning method proposed by Breiman (2001) based on bootstrap aggregation (i.e., bagging). A RF consists of multiple CARTs, with each CART grown on a bootstrap sample (i.e., sample with replacement) of the training data. Each bootstrap sample leaves out about one-third of the data, which are called the out-of-bag (*oob*) observations. The oob error is an estimate of generalization error and can be used to calculate the importance scores of input variables. To reduce correlation between trees, another design feature of RF that enhances performance is that at each split, the splitting variable is selected among a randomly chosen subset of input variables. After all the CARTs have been grown, the prediction for an unseen data point is calculated as the average of predictions from each individual CART. While being less interpretable than decision trees, RF calculates input variable importance scores that provide valuable information about the dominant factors affecting the target variable. Other popular tree ensemble algorithms include XGBoost (Chen and Guestrin, 2016) and gradient boosting machine (Friedman, 2001; Ke et al., 2017), which build the forest based on boosting algorithms.

2.1.6. Artificial neural network

Artificial neural networks (ANNs) have been widely applied to various fields including hydrology. Inspired by biological learning processes, ANNs are built out of a densely interconnected set of units. Here we briefly describe the feedforward neural networks, or multilayer perceptron networks (MLP). A typical MLP network consists of an input layer, one or more hidden layers and an output layer. Fig. 3a shows an example of an MLP with one hidden layer. For MLPs, information flows through the connections between units. Each unit, or neuron, computes a single output by passing the weighted sum of its inputs plus a bias term through a typically smooth, nonlinear activation function (e.g., sigmoid or rectifier). Using multiple hidden layers, an ANN learns a representation of the raw input, x**,** as a recursive function $f_d\left(\ldots f_j \ldots \left(f_2\left(f_1(\mathbf{x})\right)\right)\right)$, where f_j is the activation function of *j*-th layer *j* and takes a vector input (output of neurons from the prior layer) and outputs a vector (Table 2). The output layer computes the final output as the linear combination of the learned representation (the output of the last hidden layer).

The weights and biases are learned using the backpropagation algorithm. Backpropagation first evaluates the output values of each neuron in a forward pass of information. Second, it calculates the partial derivative of the loss function with respect to each learnable weight and biase. It then updates the weights and biases according to the partial derivatives in a backward pass through the layers. A hyperparameter, the *learning rate*, affects the size of the update. The process is repeated, resulting in a gradient descent approach.

ANNs are considered to have high representational power. It has been proven that a MLP with three layers can approximate any function to arbitrary accuracy given sufficient units (Cybenko, 1989; Mitchell, 1997). A major shortcoming of MLPs is that the backpropagation algorithm is only guaranteed to converge to some local minimum. Research interests in ANNs have been revived in the last decade in the context of deep learning, which is discussed in Section 2.3.

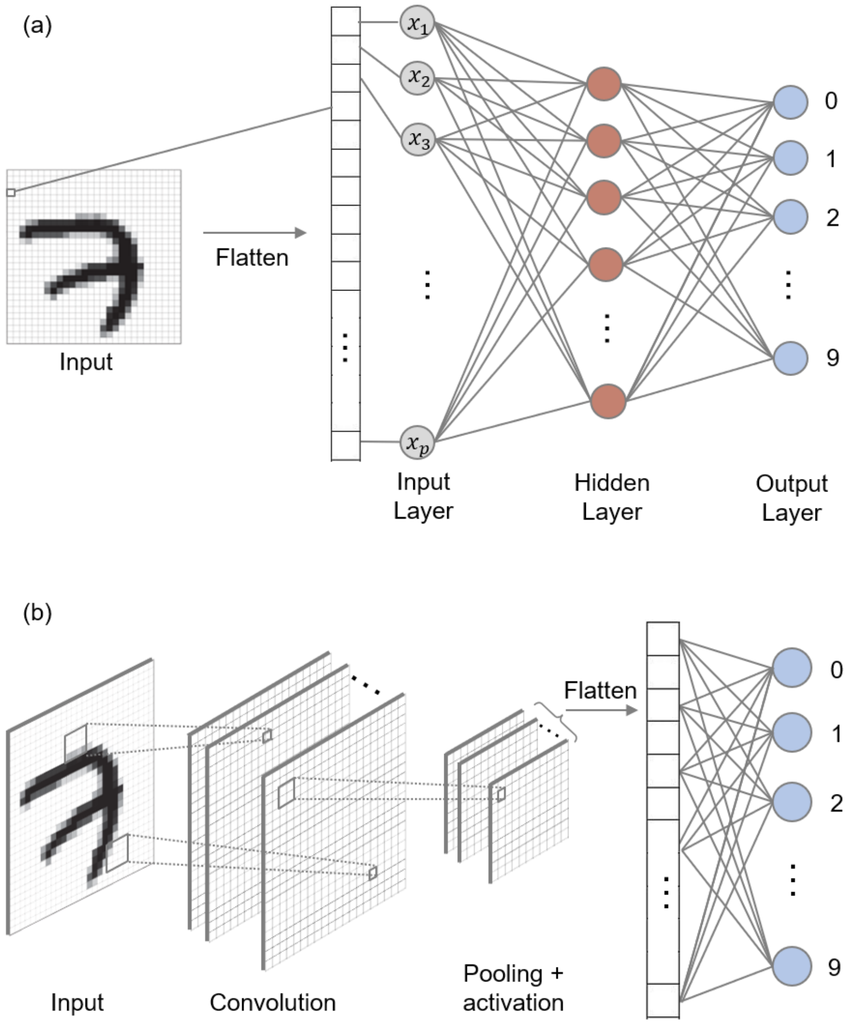


Figure 3. The architecture of (a) a fully connected ANN and (b) a CNN for classifying hand written digits. The ANN has one hidden layer, within which each neuron applies an activation function on the linear combination of inputs $\mathbf{x} = [x_1, \ldots, x_p]^T$, the flattened pixel values of the input image. The CNN applies convolution, pooling, an activation function, followed by a fully connected layer for final output (Section 2.3.2).

2.2. Model Selection

2.2.1. Comparison of machine learning algorithms

All the supervised machine learning algorithms described in Section 2.1 can be viewed as learning the target function which is a linear combination of features or representations. As summarized in Table 2, the algorithms differ at how features/representations are constructed. In the simplest case of linear regression, the raw input variables are directly used as features. Lasso goes one step further, by learning whether the coefficients are exactly zero or not. SVM and GPR use a user specified kernel (covariance) to implicitly embed the input into a higher dimensional feature space. CART learns a representation that adaptively partitions the input space into rectangular regions. The representation learned by ANN is the output from the last hidden layer, which can be written as a recursive function. Unlike the other algorithms reviewed in Section 2.1., ANN is not restricted to a particular type of representations and can automatically extract information from raw inputs. This gives ANNs and deep networks high representation power, which is further discussed in Section 2.3.1.

The choice of machine learning algorithms is often application specific. The primary decision factor is the prediction accuracy of the algorithms (generalization performance, Section 2.2.2.). Empirical studies on various benchmark datasets have suggested that tree ensemble algorithms generally work well (Fernández-Delgado et al., 2014; 2019). This is because tree-based algorithms have built-in capability of variable selection and accounting for interaction among input variables. However, many hydrologic applications involve target functions that exhibit local smoothness. In this case, it may be more advantageous to use methods such as SVM and GPR, which can enforce local smoothness by choosing an appropriate kernel (e.g., the squared exponential kernel). For applications that need to estimate uncertainty associated with the predictions, Bayesian methods such as GPR offer a natural option. Other machine learning models could use resampling methods such as bootstrapping to provide quantification of uncertainty. As will be discussed in Section 2.3.1, deep networks typically outperform conventional machine learning algorithms when dealing with unstructured data such as texts, images, and videos because of their capability of automatic representation learning.

While generalization performance is arguably the most important consideration for model selection, it is sometimes desirable to select algorithms with high interpretability. For example, Lasso produces a parsimonious linear model and is therefore easy to interpret. Besides, decision trees learn a hierarchical model structure that can be easily visualized; however, tree ensemble methods are less interpretable.

2.2.2. Generalization Performance

Generalization error, used interchangeably with *test error*, is defined as the expected prediction error, as measured by a given metric, over unseen data points, yielded by a machine learning model trained on a given training dataset. In contrast, the training error refers to the average error over the training data points. Commonly used error metrics include 0-1 loss (0 if a data point is correctly categorized and 1 otherwise) for classification and mean squared error and log likelihood for regression tasks. Because prediction is a central goal of both data-driven and process-based modeling efforts, estimating generalization error is critical for gaining confidence in a particular model for prediction tasks and selecting the best model and/or hyperparameters from a set of candidates.

Unsurprisingly, the capability of a model to fit a given set of training data increases as its complexity increases. An underfitting model will generalize poorly because it is not complex enough to capture the range of variability of the target function. For example, an ANN with 1 hidden unit will likely fit the data poorly; as more layers and hidden nodes are added to the ANN, both the training and test errors decrease because of the added representation power. However, when the model complexity exceeds the degree that can be justified by the training data, the model becomes overfitted: although training error continuously decreases, test error starts to increase (Fig. 4). An overly complex model overfits the training data in that it may extract some of the noise. Consider as an example training an ANN with M hidden units to fit n data points that follow Gaussian distribution with zero mean and unit standard deviation. When $M \geq n$, the ANN can fit the data perfectly.

However, it tends to fail at generalizing to data it has not seen before. Besides number of parameters (weights for ANNs), model complexity is also manifested by the size of the parameters. When training an ANN, it is often observed that as training epochs elapse, training error decreases as the weights are adjusted and the model gets better at fitting the training data. However, at some point the generalization error starts to increase (Prechelt, 1998).

The general trend of training and test errors can be explained by statistical learning theory. Assuming that data points in the training and test sets are independent, identically distributed, it can be shown that the training error is usually lower than the test error. The expected squared error of a trained model on an unseen data point can be decomposed into three terms. The first term is the variance of the measurement error associated with the target, representing irreducible error. The second term is the square of the bias caused by the hypothesis space of the learning method, such as approximating a nonlinear function with a linear model. The third term is the variance of the fitted model. There is usually a tradeoff between bias and variance. A more complex model yields lower bias at the expense of higher variance and thus may be prone to overfitting (Hastie et al., 2009).

In order to find the model that will yield low generalization error, the common practice is to randomly divide the dataset into training, validation, and test subsets. Shuffling is recommended so that the three subsets are approximately from the same distribution. A model is repeatedly fitted to the training set, each time using a different set of hyperparameters or machine learning algorithms. The generalization error of the fitted models will then be evaluated on the validation set. Finally, the best-performing combination of machine learning algorithm and hyperparameters is selected and evaluated with the test set.

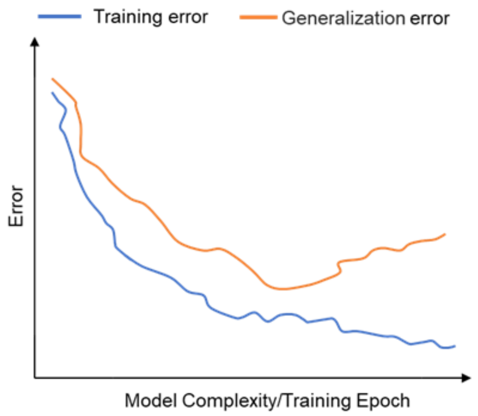


Figure 4. Schematic of trends in training and generalization errors as the model becomes more complex. When the model complexity increases, training error overall tends to decrease while test error increases, despite temporary fluctuations.

Some machine learning algorithms have their own implementations for estimating generalization error. For example, random forest uses the out-of-bag error as an estimate. Cross-validation (CV) is a model-generic approach routinely used for hyperparameter selection especially when data size is not very large. CV partitions the examples (with known inputs and target) into a training and a validation set. Multiple rounds are performed, each time using a different data partition. The resulting error metrics (e.g., misclassification rate, mean squared error) on the validation set are combined to estimate the generalization error of the model. Various implementations of CV exist, differing in how data is partitioned. Two commonly used implementations are leave-one-out (validation set consists of a single datapoint) and k-fold CV (validation set is one of k subsets).

A common practice to prevent overfitting and improve generalization performance is using regularization strategies. During training, the machine learning algorithm seeks to minimize the loss function that evaluates the misfit between the model and the given targets. For some applications, it may be desirable to impose preference to other behaviors of the learned model such as smoothness and sparsity. In order to achieve this goal, regularization techniques add a penalty to the loss function; the L_1 and L_2 norms of learned coefficients are often used as penalty, such as in Lasso and SVM, respectively. In addition to explicitly representing preference via a penalty term, regularization may be implemented implicitly. For example, the pruning technique reduces the complexity of a CART and alleviates overfitting. Training of ANNs often employs the *early stopping* strategy, which monitors the test error on a validation set and terminates the training when the test error continuously increases (Fig. 4). Regularization techniques specifically designed for deep learning will be described in Section 2.3.

2.2.3. Curse of dimensionality and variable selection

In addition to the choice of machine learning algorithms and hyperparameters, the generalization error is affected by the selection of input variables. In hydrologic applications, a variety of observed and derived data may provide some information towards the problem of interest. However, including all relevant variables pose challenges to machine learning algorithms, known as the *curse of dimensionality* (Hastie et al., 2009). Dimension reduction techniques can be used to reduce input dimensionality and improve efficiency. For example, the principal component analysis (PCA) is a commonly used dimension reduction method, which extracts linear combinations of input variables that explain most of the variability in data and then uses the combinations as inputs to machine learning algorithms. A related method, linear discriminant analysis (LDA), is a supervised dimension reduction method that takes the target variable (i.e., class labels) into consideration when extracting linear combinations of input variables (Izenman, 2013).

Dimension reduction can also be formulated as a variable selection problem, which has been studied extensively in the literature (George, 2000; Guyon and Elisseeff, 2003; Liang et al., 2008). Classical variable selection methods include backward elimination where variables are sequentially removed from the full model, forward selection where variables are sequentially added to the model, or combination of both (Blanchet et al., 2008). A variety of selection criteria can be used to determine which variable to remove or add, such as F-tests, t-test, Akaike information criterion (AIC) and Bayesian information criterion (BIC) (Burnham and Anderson, 2004). In addition to these generic methods, some supervised machine learning algorithms have built-in variable selection function. Examples include Lasso (Section 2.1.2), CART and random forests (Section 2.1.5). PCA/LDA can also be used to obtain a reduced set of input variables. Although the above-mentioned automatic variable selection techniques are powerful tools to reduce the input dimension, they should not replace careful feature selection based on expert knowledge whenever such knowledge is available.

2.3. Deep learning

2.3.1. Motivation

Conventional machine learning techniques often do not perform well for complex tasks such as computer vision, speech recognition, and natural language processing. These tasks involve large volumes of natural data in the raw form, such as images, videos and text. Consider as an example an intensively studied benchmark, the MNIST (Modified National Institute of Standards and Technology) database. The database consists of normalized grayscale scanned images of digits (0 to 9) handwritten by human individuals. When applying a conventional machine learning algorithm, the pixels within an image are typically unfolded (or flattened) into a vector, and each pixel is treated independently. An ANN can be constructed with p input units, p being the total number of pixels within an image, and multiple hidden layers. These layers are fully connected in that the learning process will attempt to learn the weights connecting each pair of units in adjacent layers (Fig. 3a), leading to a large number of learnable parameters. This greatly increases the need for training data points to make the learning problem well posed and the difficulty for an optimization algorithm to find a solution. In addition, the pixel representation of an image does not account for spatial correlation among pixels and lacks certain invariant features such as rotation and shift.

For many applications including the MNIST benchmark, careful handcrafting of features from raw data has been critical to achieve good performance with conventional machine learning algorithms. This feature engineering process relies on substantial manual efforts and domain expertise, and is application specific. When dealing with a large volume of data that have complex and nonlinear patterns, conventional machine learning with the handcrafted features is not flexible enough to extract these patterns (Najafabadi et al., 2015). Representation learning replaces manual feature engineering and automatically extracts, using a general-purpose learning procedure, representations of the raw data that might be useful for subsequent supervised learning tasks. Deep learning architectures stack multilayer neural networks to learn such representations. Each layer can be thought of as learning one aspect of the underlying structure of the data, and stacking layers composites the structures learned by individual layers. Research on deep learning theory suggests that such distributed representation endows deep learning with exponential advantages over conventional learning algorithms based on local representation (Bengio et al., 2013). It has been shown that deep networks can be efficiently trained by gradient descent methods (Rumelhart et al., 1986; Glorot et al., 2011), and greater depth generally leads to better generalization performance (Bengio et al., 2007; Ciregan et al., 2012; Goodfellow et al., 2016).

Deep learning techniques take advantage of fast GPUs and increasing data availability and have achieved record performance in various computer vision, speech recognition and natural language processing tasks. They have also been shown to hold great promise in many domains of science and engineering. In this subsection, we briefly describe some of the deep learning architectures that are the most relevant to hydrologic applications.

2.3.2. Convolutional Networks

In order to overcome the limitations of traditional ANNs on the MNIST database, LeCun et al. (1990; 1998) handcrafted neural network architecture with locally connected layers and shared weights. These neural networks significantly outperformed the fully connected ANNs on experiments centered around the MNIST database. These pioneering efforts led to the development of convolutional networks (CNNs). In 2012, a deep and wide

CNN model, AlexNet (Fig. 5, Krizhevsky et al., 2012) was proposed and won the ImageNet Large Scale Visual Recognition Challenge and outperformed all conventional machine learning and computer vision approaches. As of today, CNNs have achieved remarkable successes in computer vision and related areas. Designed for multi-dimensional arrays, CNNs use convolution operations in place of fully connected matrix multiplication. A convolutional layer applies a kernel (or filter) that calculates a local weighted sum as the kernel slides through the input array. The number of learnable weights depends only on the kernel size and is usually much smaller than the size of the input array. Multiple kernels can be applied simultaneously to output a multi-channel image (Fig. 3b). Such sparse connectivity is the key advantage of CNN over classical ANNs with full connectivity (Goodfellow et al., 2016). The local weighted sums are then passed through a nonlinear activation layer, such as ReLU that applies the rectifier activation $max(0, x)$, where x is the local weighted sum. In this way, the convolutional layer extracts local motifs of the input array or output from the previous layer. Subsequently, a pooling layer merges local features by calculating local statistics (such as max) to reduce the dimension of representation (Fig. 3b) and preserve shift invariance properties. Multiple convolutional, nonlinear, and pooling layers can be stacked (Fig. 5) to extract hierarchical patterns where higher-level features are derived by composing lower-level features (LeCun et al., 2015). Finally, the high-level features are usually flattened before passing through a fully connected layer for classification or regression (Fig. 3b and 5).

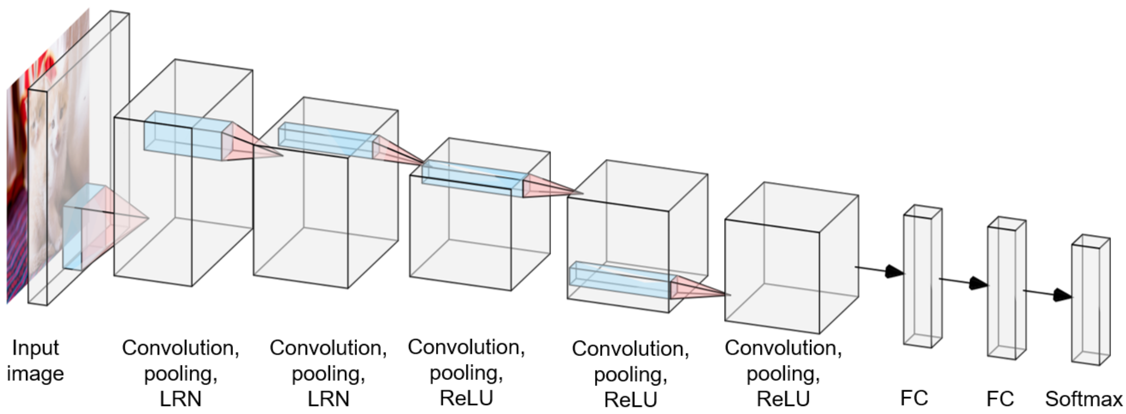


Figure 5. The architecture of the AlexNet (Krizhevsky et al., 2012) consists of convolution, max-pooling, local response normalization (LRN), ReLU and fully connected (FC) layers.

2.3.3. Recurrent Neural Networks for Sequence Modeling

Recurrent Neural Networks (RNNs) are designed for modeling sequential data such as time series with some underlying temporal dynamics. An RNN digests one element (e.g., a word, streamflow at one time step) of the input sequence at a time and uses its hidden units to keep information learned from the past elements of the sequence. Therefore, we can “unroll” the RNN and consider it as a chain of recurrent neurons, each corresponding to one time step (Fig. 6). Similarly to the sparse connectivity of CNNs (i.e., sharing weights across different locations of the input multidimensional array), RNNs share weights across different locations (in time) in the input sequence. While the RNN architecture can represent complex dynamics, its training suffers from the well-known vanishing gradient problem. The backpropagated gradients either grow or shrink at each time step; after many time steps, the gradients will either explode (leading to unstable optimization) or, more likely, vanish. Almost zero gradients greatly slow down the learning process because each iteration would apply a very small update to the weights (Bengio et al., 1994; Hochreiter, 1998).

Long-short term memory (LSTM) is an RNN architecture proposed to overcome the vanishing gradient problem. LSTM and its variants have proven powerful for learning long-term dependencies in time series (Graves, 2012; Greff et al., 2017). Each LSTM cell corresponds to one time step, repeats to form *N* recurrent layers, and retains past information in cell memory. Fig. 6 shows the classical LSTM architecture (Hochreiter and Schmidhuber, 1997). At each time step t, the current input x_t is combined with hidden state (h_{t-1}) and cell memory (c_{t-1}) from the previous time step to determine whether the input will be accumulated to cell memory c_t according to the input gate i_t and whether the past cell memory c_{t-1} will be forgotten according to the forget gate f_t. The output gate o_t then determines whether the hidden state h_t will be updated with the cell memory c_t.

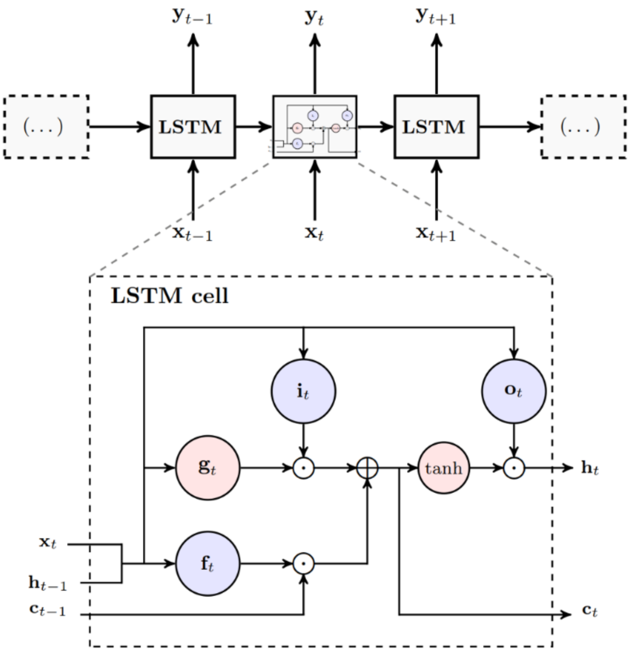


Figure 6. A recurrent neural network (RNN) with LSTM cells. At time step t, x_t is the current input, c_t is the cell memory, h_t is hidden state, i_t, f_t, o_t are the input, forget, and output gates, respectively, g_t is the cell input activation vector, and ⊙ denotes element-wise array multiplication.

2.3.4. Other popular architectures

Representation learning techniques are capable of automatically learning representations of the raw input, thus providing insights into the data and/or help with the subsequent supervised learning (Bengio et al., 2013). Examples include K-means that learns representations as the centroid of clusters, PCA that generates eigenvectors as a linear representation, and convolutional and pooling used in a CNN that learn motifs in the input image. In addition to these techniques, autoencoders are an important type of deep learning architecture for representation learning (Goodfellow et al., 2016). An autoencoder attempts to learn a low dimensional representation of the data. A simple autoencoder consists of an input layer, a hidden layer, and an output layer. The sizes of input and output layers are equal to the size of the input, while the hidden layer is typically smaller. As a result, the autoencoder must learn to compress information (encode) in the input and then reconstruct the input from the compressed representation stored in the hidden layer (decode). Further, we can impose desired properties on the learned representation, such as sparsity (sparse autoencoder) and robustness to noise (denoising autoencoder); these regularized autoencoders have proven effective in learning representations helpful for subsequent classification tasks (Vincent et al., 2010). Recently, several Bayesian autoencoders have been proposed, known as variational autoencoders, since variational algorithms are used to learn the probabilistic description of the latent representation (Kingma and Welling, 2014; Sønderby et al., 2016). In the Bayesian version of autoencoders, the encoder produces the (approximated) posterior distribution of the latent representation, and the decoder samples one or more realizations from the estimated posterior to generate reconstructions of the original input.

Generative adversarial network (GAN) is another architecture for generative learning. GAN learns to generate new data with the same statistics as a given training set (usually images). A generative network and a discriminator compete with each other in the form of a zero-sum game (Goodfellow et al., 2014; Creswell et al., 2018). The generative network, typically based on deconvolutional layers, synthesizes candidates that are similar to the training data with the objective to "fool" the discriminator network, while the discriminator attempts to distinguish synthesized candidates from the true data. Through this process, the GAN gets better at generating synthetic data that resemble the training data. Because the generative network is implicitly trained through the discriminator, and the discriminator is being updated, GAN is particularly suitable for unsupervised learning although it can also be used for supervised and semi-supervised learning where training data are scarce. GANs have attracted wide attention due to potential use for malicious applications such as producing fake photographs and videos. As discussed in Section 3.2.1, GANs have important applications in inverse modeling of geologic media.

Finally, in recent years *attention* has become a very influential idea in the deep learning community. Attention enables a deep network to focus on certain parts of the input data in a way similar to how human beings would pay attention to different regions of an image or correlate words at different locations in sentences. This is achieved through learning importance weights that describe how strongly the target is correlated to the elements of input data. There are various attention mechanisms designed to accompany CNNs, RNNs and other architectures. They have achieved high performance for many tasks such as image captioning (Vinyals et al., 2015) and translation (Vaswani et al., 2017; Chaudhari et al., 2020).

2.3.5. Common practices and other considerations

Learning the weights for a deep network is usually a hard problem, and standard gradient descent and random initialization often perform poorly (Glorot and Bengio, 2010). As a result, various initialization strategies and variants of gradient descent have been proposed (e.g., Bottou, 2010; Saxe et al., 2011; Sutskever et al., 2013; Kingma and Ba, 2015). Because deep learning often deals with very large amounts of data posing computational challenges, a common practice is to divide the datasets into small subsets, called a *mini-batch*. At each iteration, a mini-batch is loaded and backpropagation is executed, leading to mini-batch gradient descent (Li et al., 2014). This is repeated until all mini-batches have been used, concluding one *epoch*. The training process lasts for multiple epochs; the number of epochs is a user-specified parameter but may be determined using the early stopping strategy. Learning rate plays an important role in the training and generalization performance of deep networks. At the simplest form it can be specified as a constant hyperparameter. A number of methods have been developed recently that adapt the learning rates and training progresses, such as Adam (Kingma and Ba, 2015).

The regularization strategies for conventional machine learning algorithms discussed in Section 2.2.2 mostly apply to deep learning as well. In addition to those strategies, *dropout* (Srivastava et al., 2014) is a computationally efficient and powerful method specifically designed for deep learning. Dropout can be thought of as a practical approximation to the idea of bagging in ensemble learning (such as the random forest). Traditional bagging requires training and retaining multiple models and would become computationally unaffordable for very large neural networks. Dropout omits a portion (as determined by dropout rate) of the weights during training, thus regularizing the complexity (and variance) of the learned network. More precisely, each time a mini-batch is loaded, only the weights of a randomly selected subset of the neurons will be updated by backpropagation. The added cost of applying dropout at each step to a specific network is negligible. It was shown that dropout is more effective than other regularization methods including L_1 and L_2-norm based (Srivastava et al., 2014).

Hyperparameters such as learning rate and dropout rate typically need to be tuned to improve generalization performance. Methods such as grid-search work well for conventional machine learning methods but may become computationally expensive for deep learning. For an overview of automatic hyperparameter optimization algorithms and general recommendations for manual tuning, readers are referred to Goodfellow et al. (2016) and Hutter et al. (2019).

3. APPLICATIONS IN HYDROLOGIC SCIENCES

3.1. Machine Learning as a Stand-alone Model

3.1.1. Detecting patterns and events from remote sensing data

The recent growth in hydrologic data volume has been boosted largely by increasing availability of remote sensing data. Remote sensing provides measurements directly or indirectly related to the water cycle with unprecedented spatial coverage. While some products have been available for decades, recently remote sensing is increasingly used as more products become available and cyberinfrastructure advances lower the barriers to accessing and using these data. Particularly in areas where *in situ* monitoring networks are sparse or missing, remotely sensed data are an important source of information for large scale monitoring of patterns and events related to hydrologic sciences as well as estimating key hydrologic variables (Fig. 7). This section briefly reviews applications in which machine learning is used for classification; regression applications will be discussed in Section 3.1.2.

Machine learning is being used to identify water-related land cover changes and land surface features from remote sensed data, often leveraging cloud computing platforms (e.g., Google Earth Engine, Gorelick et al., 2017) to process large quantities of geospatial data (e.g., Deines et al., 2017; Gao et al., 2018; Cho et al., 2019; Yuan et al., 2020 and references therein). For example, Deines et al. (2017) used a random forest classifier to identify irrigated areas in the High Plains, an arid to semi-arid region, based on high resolution multi-spectral satellite imagery. In another study, a set of novel input features, such as weather sensitive remote sensing indices of a sub-humid area, were hand crafted to enhance the contrast between neighboring rainfed and irrigated areas; these features then enabled a random forest classifier to achieve satisfactory performance in mapping irrigated areas (Xu et al., 2019, Fig. 8). This type of application often has a large number of potential input variables with high correlation among some of the inputs. Random forest automatically performs feature selection and is robust when collinearity exists, making it particularly suitable for this and similar applications. On the other hand, deep learning algorithms may be promising alternatives for bypassing feature engineering efforts. Deep learning was recently applied in climate science to detection of extreme weather events such as tropical cyclones, atmospheric rivers and weather fronts. Detecting such extremes have traditionally relied on human expertise and subjective detection thresholds. As introduced in Section 2.3.2, convolutional layers can automatically extract patterns from image-like data, making them suitable for climate pattern identification from massive climate datasets (Liu et al., 2016; Racah et al., 2017; Kim et al., 2019).

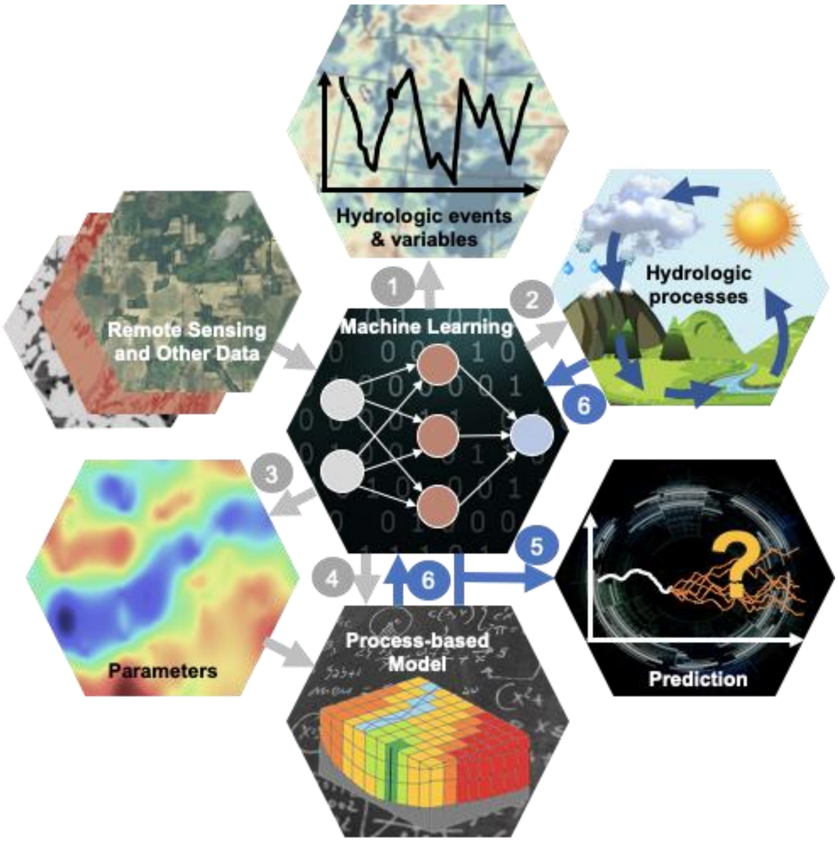


Figure 7. Machine learning has been used in various hydrologic applications in stand-alone mode or integrated with process-based modeling. Machine learning can process multi-type data to identify hydrologic events and estimate variables (1), approximate hydrologic processes and generate new knowledge regarding the processes (2), aid in parameterization of process-based models, develop fast surrogates (4), and correct the bias of process-based models (5). The current research frontier is to explore hybrid modeling that integrates physical knowledge with machine learning to achieve improved prediction accuracy and interpretability (5, 6) (Karpatne et al., 2019; Reichstein et al., 2019). Arrows indicate information flow.

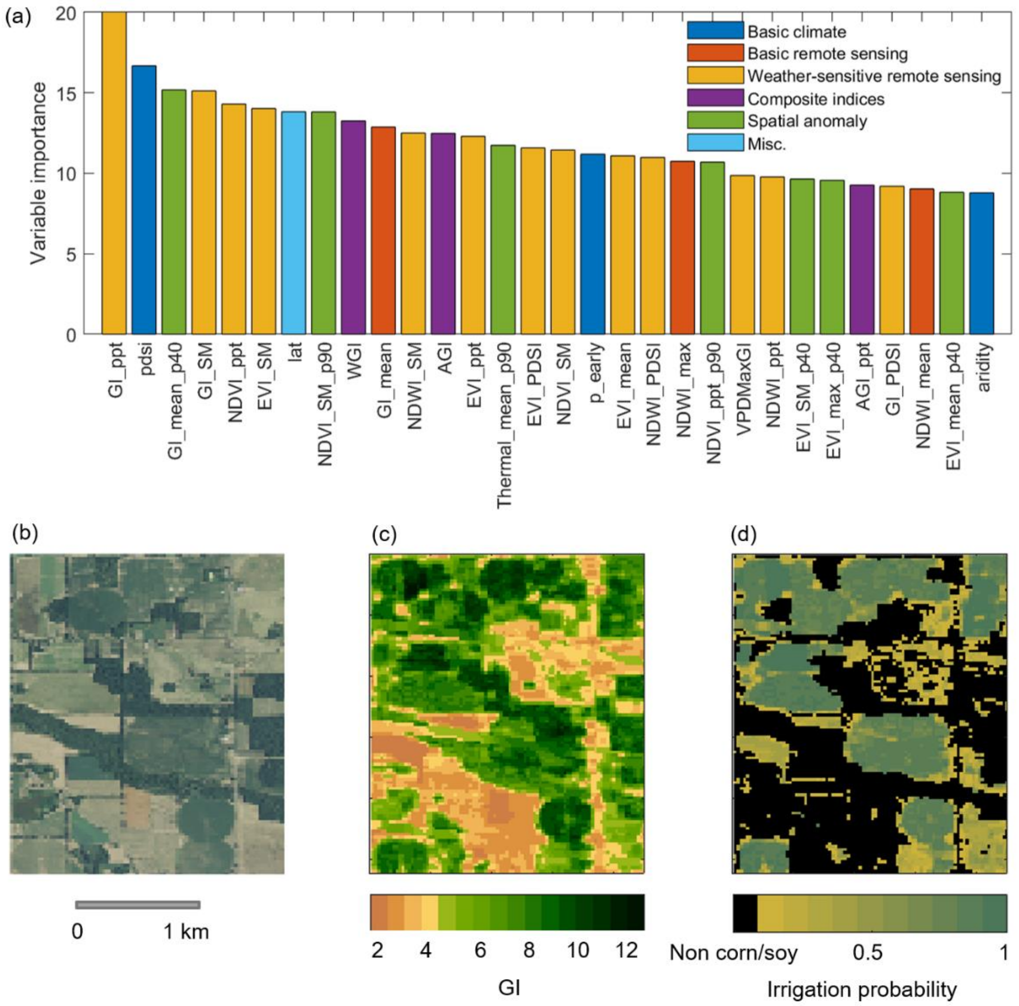


Figure 8. A random forest (RF) classifier was developed to map irrigated fields at 30 m resolution for a subhumid temperate region. (a) Top 30 (out of 98) important features as identified by RF. Different colors indicate categories of features, such as weather-sensitive remote sensing indices. (b) National Agriculture Imagery Program (NAIP) aerial image showing irrigated farms with varying sizes. NAIP is shown for visual comparison and not used by the RF classifier. (c) Weather-sensitive GI calculated from remote sensing images that immediately followed a dry period. (d) Segment of irrigation probability map generated by RF for 2012. Areas not classified as corn or soybeans are shown in dark. Recreated from Xu et al. (2019) under Creative Common CC BY License.

3.1.2. Estimating hydrologic variables

Hydrologic variables such as precipitation, snow water equivalent (SWE), evapotranspiration (ET), and soil moisture often exhibit high spatial and temporal variability. Remote sensing products provide valuable information regarding the variability of these variables where ground stations do not exist or are sparse. Because these hydrologic variables are not directly measured by the payload onboard a satellite or UAV, they are usually estimated based on a presumed relationship between the variable and signals collected by the payload and covariates. Machine learning algorithms are powerful tools for this purpose because they can easily incorporate various types of input data without resorting to presumed relationships. In particular, GPR is a popular choice because it can enforce local smoothness, which is often desirable for hydrologic variables.

Estimation of precipitation is critical for climatic and hydrologic research. PERSIANN and its variants are arguably the most successful machine learning-derived, remote sensing-based precipitation estimates (Sorooshian et al., 2000; Ashouri et al., 2015; Tao et al., 2016). Earlier versions of PERSIANN used the classical ANN to estimate precipitation from satellite longwave infrared imagery. Recently, Tao et al. (2016) used a stacked denoising autoencoder to improve estimation accuracy; the deep network was shown as able to substantially alleviate bias and false alarms. A follow-up study combined PERSIANN precipitation with LSTM to provide short-term precipitation forecast (Akbari Asanjan et al., 2018). Motivated by the spatiotemporal correlation structure underlying the precipitation field, the convolutional layer and LSTM architectures have been combined and applied to precipitation nowcasting from radar data (Shi et al., 2015; Shi et al., 2017). Conventional machine learning and deep learning methods have also been used for statistical downscaling and merging spaceborne, ground-based, and rain gauge precipitation measurements (Kleiber et al., 2012; Chen, H. et al., 2019; Pan et al., 2019; Vandal et al., 2019).

Machine learning methods have been used to estimate SWE (Bair et al., 2018; Broxton et al., 2019), ET (e.g., Ke et al., 2016; Xu, T. R. et al., 2018) and soil moisture (e.g., Ahmad et al., 2010; Zhang et al., 2017; Aboutalebi et al., 2019; Lee et al., 2019) from remote sensing and *in situ* measurements. For example, Bair et al. (2018) estimated SWE in the watersheds of Afghanistan in real time using physiographic and remote sensing data. Ke et al. (2016) used machine learning and 30-m resolution Landsat imagery to downscale MODIS 1-km ET. Aboutalebi et al. (2019) estimated moisture content of different soil layers from high-resolution UAV multi-spectral imagery and compared the performance of genetic programming (a combination of an evolutionary algorithm and artificial intelligence), ANN, and SVM. They found that the performance of machine learning algorithms increases for deeper soils, and that genetic programming achieved significantly higher accuracy than SVM and ANN at the deepest validation point. In addition, genetic programming outputs an equation that can be potentially transferred to other regions. At a larger scale, Zhang et al. (2017) used deep learning to estimate soil moisture for all croplands of China from Visible Infrared Imaging Radiometer Suite (VIIRS) raw data. Assessed using *in situ* measurements, the estimated soil moisture was more accurate than the Soil Moisture Active Passive (SMAP) active radar soil moisture and the Global Land Data Assimilation System (GLDAS) products. In addition to remotely sensed data, machine learning algorithms can also be used to leverage *in situ* moisture measurements. For example, Andugula et al. (2017) used GPR to upscale point-based soil moisture measurements from a dense sensor network.

In groundwater hydrology, there are emerging applications of machine learning. Seyoum et al. (2019) estimates groundwater level anomaly by downscaling GRACE Terrestrial Water Storage Anomaly (TWSA). Smith and Majumdar (2020) used random forests to map land subsidence due to groundwater pumping based on ET, land use, and sediment thickness. Various studies have illustrated the use of conventional machine learning algorithms to map groundwater potential based on topographic, land use, and geologic factors (e.g., Naghibi et al., 2017; Chen et al., 2019; Kordestani et al., 2019). The mapping accuracy was found sensitive to the size of the training dataset (Moghaddam, D.D. et al, 2020). Moghaddam, M.A. et al. (2020) estimated the flux between a river and groundwater from high frequency observations of subsurface pressure and temperature using CART and gradient boosting.

In addition to the above studies, machine learning has been used in environmental monitoring applications such as predicting recreational water quality advisories (Brooks et al., 2016), estimating groundwater nitrate concentration (Nolan et al., 2015), and identifying facilities likely to violate environmental regulations (Hino et al., 2018).

3.1.3. Approximating hydrologic processes

Various studies have used machine learning to model hydrologic processes such as runoff generation. Rainfall-runoff modeling and streamflow forecasting have profound implications for water resources management and have been investigated for decades. Applications of machine learning to rainfall-runoff modeling can be dated back to the 1990s (Buch et al., 1993; Kang et al., 1993; Hsu et al., 1995; Smith and Eli, 1995). While the earliest applications were focused on ANNs, later studies have employed a variety of conventional machine learning algorithms (Yaseen et al., 2015 and references therein), such as SVM (Asefa et al, 2006; Rasouli et al., 2012; Adnan et al., 2020), GPR (Rasouli et al., 2012), multivariate adaptive regression splines (Adnan et al., 2020), and ANN-based methods (Rasouli et al., 2012; Ren et al., 2018; Boucher et al., 2020). There is no consensus on a single machine learning algorithm that outperforms others; in many applications they achieved satisfactory results at various time and spatial scales and across different hydrologic regimes.

Conventional machine learning algorithms, with the exception of autoregressive models, do not have mechanisms to explicitly represent the temporal evolution of the hydrologic processes. Therefore, applying conventional machine learning to rainfall-runoff modeling requires hand-crafting a set of input features that encapsulate some “history” of the watershed, such as lagged meteorological time series. Recently, there has been a growing interest in applying RNNs, LSTM in particular, to rainfall-runoff modeling and streamflow forecasting because these deep learning architectures can represent long-term dependencies (Kratzert et al., 2018; Kratzert et al., 2019b; Jiang et al., 2020; Tenant et al., 2020). For example, Kratzert et al. (2018) used LSTM to simulate daily streamflow using meteorological forcings including daily precipitation, maximum and minimum temperature, shortwave downward radiation, and humidity. It was shown for some watersheds that the LSTM was able to use its cell memory to approximate the watershed storage dynamics such as snow accumulation and melt within the annual cycle. This likely explains the superior performance of LSTM over RNN (Fig. 9). In addition, it was found that LSTM achieved overall good performance as a regional model when it was trained using data from many catchments. When the regional LSTM model was fine tuned for individual catchment separately, it outperformed a commonly used hydrologic model (SAC-SMA combined with Snow-17) calibrated for individual catchments in the CAMELS dataset. A follow-up study further investigated the capability of LSTM as a regional model and modified the vanilla LSTM architecture to embed catchment characteristics as static inputs in addition to time-varying meteorological forcings (Kratzert et al., 2019b). The resulting LSTM model outperformed several lumped and distributed hydrological models. Besides rainfall-runoff modeling, LSTM has been used for short-term flood forecasting with lead time of hours to days (e.g., Hu et al., 2019; Lv et al., 2020; Xiang et al., 2020). For example, Hu et al. (2019) developed a spatio-temporal flood forecasting framework where proper orthogonal decomposition and SVD were applied to reduce the dimension of the large training data and the computational cost associated with training and forward evaluation of the LSTM model. Ding et al. (2019) combined attention mechanisms with LSTM; the resulting model outperformed LSTM without attention, SVM, and ANN. Besides LSTM, other deep learning architectures such as autoencoders have also been used for streamflow forecasting (Liu et al., 2017).

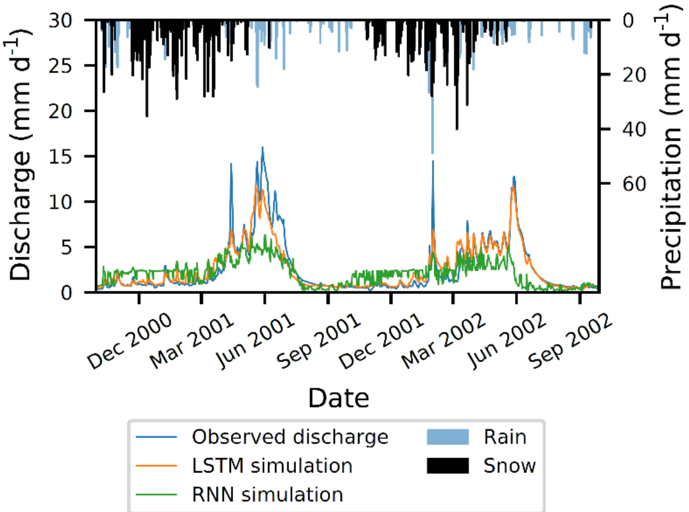


Figure 9. Observed and simulated daily streamflow at USGS Gage 13340600 for two water years. LSTM outperformed RNN during the validation period. Precipitation is partitioned into rain or snow based on minimum temperature being above or below zero. Adapted from Kratzert et al. (2018) under Creative Commons Attribution License.

Machine learning algorithms have been used to emulate dynamic processes that govern key hydrologic variables including ET and soil moisture (e.g., Torres-Rua et al., 2011; Fang et al.; 2017; Zhao et al., 2019; Fang and Shen, 2020). Torres-Rua et al. (2011) used the relevance vector machine algorithm to forecast daily PET under limited climate data conditions. Zhao et al. (2019) developed a physics-constrained RNN model to predict ET by embedding surface energy conservation into the loss function. Fang et al. (2017) used an LSTM to reproduce SMAP surface soil moisture content product over CONUS. An LSTM was trained using the SMAP product as the target, and meteorological forcings and outputs from land surface models were used as inputs. The LSTM model was able to reproduce the soil moisture dynamics with higher accuracy than regularized linear regression, autoregression, and a simple ANN.

In the groundwater hydrology community, there is also a growing body of research applying machine learning techniques. Some of these studies are focused on predicting groundwater level from meteorological variables using conventional machine learning (Yoon et al., 2011; Sahoo et al., 2017; Wunsch et al., 2018; Guzman et al., 2019) and deep learning (Ghose et al., 2018; Zhang et al., 2018; Ma et al., 2020). Other studies have investigated the potential of machine learning for groundwater flow simulation. Because training data is often scarce for this type of applications, physical constraints have been found useful. Tartakovsky et al. (2020) used fully connected DNNs for steady state saturated and unsaturated flow. The DNNs were trained to approximate the hydraulic conductivity and spatially varying state variables (head for saturated flow and pressure for unsaturated flow) with sparse observations. Physical constraints were introduced by adding the residual of the governing equation (Darcy's Law/Richards equation) to the loss function. The approach was tested on synthetic case studies and achieved satisfactory accuracy of simulating the head-conductivity relationships. Wang et al. (2020) used a similar approach for transient saturated flow simulation and added the residuals of both the governing equation and boundary conditions to the loss function. The physically constrained DNN yielded a more physically feasible solution and lower generalization error than a DNN without these constraints.

3.1.4. Mining relationships among hydrologic variables for knowledge discovery

Disentangling the interactions among multiple variables is important for understanding the dynamic behavior of the water systems. The increasing volume of observations provides opportunities for using data-driven techniques to identify the relationships among hydrologic variables without relying on physical knowledge. For example, Goodwell and Kumar (2017) used metrics based on information theory to unravel forcing and feedback relations in an ecohydrological system using high frequency data from a flux tower. Zeng et al. (2017) used SVM to analyze the competitive or complementary relationship between reservoir operation decisions for hydroelectricity production and water releases for irrigation. Another potential venue of applying machine learning for knowledge discovery is mining relations that cannot be modelled from a physical process-based perspective such as the two-way feedback between human and water systems (Pande and Sivapalan, 2017; Meempatta et al., 2019). Interpretable machine learning algorithms such as tree-based methods and Lasso hold promise for this purpose because the learned models can be interpreted to derive rules or functional relationships. For example, Hu et al. (2017) used directed information graphs and boosted regression trees to derive rules of farmers' pumping behavior in a case study in the US Midwest. In addition, the successes big data and deep learning have achieved in predicting human behavior (e.g., Van den Oord et al., 2013; Elkahky et al., 2015; Phan et al., 2017; Sohangir et al., 2018) suggest they could be promising tools to model human decision making such as irrigation and adaptation to global change.

3.2. Integration of Machine Learning with Process-based Modeling

Physical process-based numerical models have long been the primary quantitative tools in hydrologic sciences. Here we briefly review usage of machine learning integrated with process-based modeling to facilitate or improve one or more components of the latter (Fig. 7).

3.2.1. Parameterization

Most process-based models require specification of parameters. Often, the parameters do not correspond to directly measurable quantities, or it is infeasible to measure these quantities at the spatial resolution and scale required by the model. In recent years, deep learning in particular has been used to estimate properties of geologic media, such as permeability and diffusivity directly from micro-CT images of porous media (Kamrava et al., 2020; Wu et al., 2018; Wu et al., 2019). For example, Wu et al. (2018) demonstrated the utility of a physics-informed deep network for fast prediction of permeability directly from images. They first generated images of synthetic porous media, and then performed lattice Boltzmann simulations to calculate the permeability of each sample image. This resulted in a dataset that was used to train a modified CNN. The convolutional layers extract latent features from the image that could be relevant to permeability; an MLP then digests the extracted features along with two physical parameters, porosity and specific surface area, to estimate permeability. The physics-informed CNN achieved high test accuracy and outperformed regular CNN without physical parameters. Because fluid dynamics simulations such as lattice Boltzmann are computationally expensive, once trained the deep network can greatly reduce the computational cost for predicting permeability of a new image.

Generative deep learning architectures such as GANs and variational autoencoders are capable of generating data that preserve some desired properties. They are well suited for reconstruction of geologic media, often in order to generate realizations for subsequent stochastic simulations in subsurface hydrology. Laloy et al. (2017) used the variational autoencoder to construct a low-dimensional latent representation of complex binary geologic media with a relatively low number of parameters, thus making it possible to perform time consuming Markov Chain Monte Carlo (MCMC) sampling. The autoencoder outperformed the state-of-the-art inversion technique using multi-point statistics and sequential geostatistics simulation. They noted, however, that the variational autoencoder model requires several tens of thousands of training images. A follow-up study (Laloy et al., 2018) used GANs to replace the variational autoencoder in order to reduce training data needs and extend to multicategorical data (geologic facies).

In surface hydrology, machine learning has been used for regionalization of rainfall-runoff model parameters, which is an important step towards runoff prediction in ungauged basins (Beck et al., 2016; Jiang et al., 2020). For example, Beck et al. (2016) developed global maps of parameters for a simple conceptual rainfall-runoff model based on climatic and physiographic factors, using a model trained on calibrated parameters from more than 1,700 catchments. A related line of research used streamflow signatures to delineate catchments groups with distinct hydrological behaviors, wherein clustering analysis and decision trees were used for this purpose (e.g., Toth, 2013; Sawicz et al., 2014; Boscarello et al., 2016). Chaney et al. (2016) used random forest to develop probabilistic estimates of soil properties at 30-m resolution for CONUS based on geospatial environmental covariates such as distribution of uranium, thorium, and potassium.

3.2.2. Surrogate modeling

Recently, there has been increasing interest in the use of machine learning for surrogate modeling for optimization (Asefa et al., 2005; Cai et al., 2015; Wang et al., 2014; Wu et al., 2015) and uncertainty quantification (Xu et al., 2017; Yang et al., 2018; Zhang et al., 2020). Recent studies have also used deep learning for uncertainty quantification (Hu et al., 2019; Laloy and Jacques, 2019; Mo et al., 2019a; 2019b). Many process-based models, such as groundwater flow and solute transport models, are computationally expensive, making it challenging to perform analyses that require running the model for many times (Asher et al., 2015). Surrogate models emulate process-based model simulation results as a function of inputs and/or parameters but run much faster. Machine learning techniques are powerful tools to represent nonlinear functions and thus well positioned for surrogate modeling. For example, Cai et al. (2015) used SVM to develop a fast surrogate of a watershed simulation model (SWAT); the surrogate model was coupled with a stochastic optimization model within a decision-support framework to assess the roles of strategic measures and tactical measures in drought preparedness and mitigation under different climate projections. Wu et al. (2015) used an adaptive approach, where the surrogate model is adaptively refined during the search for optima. Xu et al. (2017) used random forest and

SVM to construct fast surrogates of a regional groundwater flow model for Bayesian calibration. Mo et al. (2019a; 2019b) used a convolutional encoder-decoder architecture to build surrogate models to facilitate groundwater contaminant source identification and uncertainty quantification of a multiphase flow problem, respectively. Laloy and Jacques (2019) compared three surrogate modeling techniques (GPR, polynomial chaos expansion, and DNN) for sensitivity analysis and Bayesian calibration of a reactive transport model. DNN achieved the best emulation accuracy even though the training set is relatively small (from 75 to 500 samples). However, the DNN surrogate model yielded the worst performance for the calibration task and led to posterior distribution far away from the truth. A possible cause is DNN overfitting the training data, resulting in small but biased prediction error with a complex structure. In contrast, GPR-based surrogate model approximated the true posterior well. The findings suggest the need for further investigation on quantification of uncertainty introduced by surrogate modeling. Zhang et al. (2020) used GPR and PCE to construct surrogates for Bayesian calibration of a groundwater transport model. They adaptively refined the surrogates, thus reducing surrogate error, as the posterior distribution is being approximated. For uncertainty quantification, GPR is a convenient choice since it naturally fits into the Bayesian framework (Kennedy and O'Hagan, 2001). In addition, GPR can enforce local smoothness, which may be beneficial for parameter estimation and optimization (Razavi and Tolson, 2013; Laloy and Jacques, 2019).

3.2.3. Bias correction

Process-based models are generally considered more reliable than machine learning-based data-driven models for predictive tasks such as projection under climate change. However, it has been recognized that process-based models may yield biased simulation results due to errors in forcing data, incorrect parameters, and/or simplified or improper conceptualization of the physical processes despite advances in understanding of hydrologic processes and development of sophisticated model structures (Liu and Gupta, 2007; Demissie et al., 2015; Xu et al., 2017). Machine learning techniques may be able to learn from observational data to recover information not represented by process-based models. Because process-based and data-driven modeling have complementary strengths, they can be combined to yield more accurate predictions. Conventional machine learning techniques have proven effective in correcting the bias of surface (Abebe and Price, 2003; Solomatine and Shrestha, 2009; Pianosi et al., 2012; Evin et al., 2014 and references therein) and subsurface hydrologic models (Demissie et al., 2009; Xu et al., 2015; Tyralis et al., 2019). Recently, there is emerging research applying deep learning for bias correction. Sun et al. (2019) used CNN to correct the mismatch between NOAH-simulated terrestrial water storage anomaly (TWSA) and GRACE products. Nearing et al. (2020) used LSTM to process the output of a calibrated conceptual rainfall-runoff model and achieved better accuracy than using each model alone. Frame et al. (2020) applied a similar approach to post-process the daily streamflow predictions of the National Water Model (NWM), leading to substantial improvements. The LSTM performance increased when NWM states and fluxes were added as inputs.

4. CHALLENGES AND OPPORTUNITIES

In the past, application of machine learning in hydrology and other disciplines of geosciences had been largely hindered by three primary challenges. These challenges include possible degradation of generalization error, the lack of physical interpretability and constraints, and small sample size. Even with regularization strategies implemented, a trained machine learning model may still generalize poorly. This issue is exacerbated by the relatively small training dataset available in hydrologic applications as well as the need to predict under nonstationary conditions such as those induced by climate change. Hydrologic applications are also known to exhibit high degrees of spatial heterogeneity. Most previous applications of machine learning in hydrology are limited to one or a few test cases, and the machine learning models developed for a limited number of sites are likely not transferable to other regions where training data is scarce. Although the extrapolation problem exists even for process-based models, it is particularly acute for machine learning methods partly because of their flexibility of adapting to a wide range of functional relationships and lack of physical constraints. In addition, machine learning may also fall short of predicting emerging patterns.

A second major challenge lies in the lack of physical interpretability of machine learning models. With few exceptions (e.g., Lasso, CART), most machine learning models learn functional relationships that are very complicated to comprehend. It is usually difficult, if at all possible, to draw physical understanding from the learned model. In addition to the models themselves being hard to interpret, they may provide predictions that cannot be easily understood, are implausible, and/or lack physical consistency. The lack of transparency raises questions about the appropriateness of using machine learning models for decision making that has high stakes.

Because of this and also given the importance of knowledge discovery in any discipline of physical sciences, developing approaches to probe into these models and inherently interpretable machine learning models is crucial. In recent years, there has been a surge of work on the topic of "explainable AI" within the deep learning community (see Gilpin et al., 2018; Rudin et al., 2019; Samek and Müller, 2019 and references therein). In the hydrology community, interpreting deep learning models is also gaining attention (Shen, 2018; Ding et al. 2019; Kratzert et al., 2019a).

A current research frontier is to integrate knowledge about physical processes with machine learning. Process-based modeling and data-driven modeling have complementary strengths and weaknesses, and combining them in multiple ways provide exciting opportunities to address the above-mentioned challenges. Karpatne et al. (2017) and Reichstein et al. (2019) provide comprehensive recommendations on possible ways physical knowledge and machine learning can be integrated. Here, we highlight a few integration mechanisms that have proven to be promising in hydrologic applications. First, physical knowledge can be incorporated as regularization terms in the loss function. In this way, the learned model is forced to respect physical constraints such as mass and energy conservation (de Bezenac, 2019; Jia et al., 2019; Tartakovsky et al., 2020; Wang et al., 2020). Second, a hybrid model can consist of a process-based component responsible for physical processes that are well understood and a machine learning component dealing with the less understood processes (Ren et al., 2018; Sun et al., 2019). In some cases, it may be possible to encode the physical knowledge expressed as ordinary or partial differential equations into the deep learning architecture (Jiang et al., 2020). When explicit encoding is not possible, an alternative is to augment training data of the machine learning model with simulation results generated by a process-based model (Jia et al., 2019). This provides two-fold benefits: more training data and the potential to learn physical knowledge, potentially related to predicting under nonstationary conditions, from the augmented training data. It has been shown in some studies discussed above and reviewed in Section 3 that incorporating physical knowledge improves the generalization performance of the machine learning model.

A third challenge arises from small sample size in hydrologic applications. Despite the fast-growing hydrologic data availability, data are still scarce in some applications, especially when data are expensive or time-consuming to collect. For example, there may be a limited amount of ground truth of the output variable, or available training data may have imbalanced classes due to sampling bias or the output variable of interest being a low probability event (e.g., Deines et al., 2017; Xu et al., 2019). In addition, information does not necessarily increase linearly with data amount. For example, one year of streamflow observations at 15-min interval (~35,040 data points) is likely insufficient to properly train a machine learning model for rainfall-runoff modeling due to autocorrelation and the limited range of the hydrologic regime the training data covers. The importance of the "informativeness" of the data (Gupta et al., 1998) has been investigated in various studies both theoretically (Gupta and Sorooshian, 1985) and empirically (Yapo et al., 1996; Boughton, 2007; Singh and Bárdossy, 2012). These studies provide valuable insights into determining the amount of data needed to train machine learning models in hydrologic context. Ayzel and Heistermann (2021) train deep learning-based rainfall-runoff models for six CAMELS watersheds using varying data length and found that deep learning models require longer data to calibrate than a conceptual hydrologic model, although their performance catches up quickly with increasing data length. Their findings suggest that in practice it may require less data to train the deep learning architectures than predicted by theoretical bounds of sample size established in deep learning literature (e.g., Du et al., 2018). Problems associated with small sample size may be alleviated by the above-mentioned physics-informed machine learning methods and borrowing ideas from unsupervised learning, semi-supervised learning (Zhu and Goldberg, 2009; Kingma et al., 2014; Ding et al., 2018) or active learning (Settles, 2011) to utilize available data more efficiently (Racah et al., 2017; Karpatne et al., 2019).

Related to the problem of small sample size is the juxtaposition of multi-source, multi-type, multi-scale data with various accuracy. Machine learning algorithms do not have a mechanism to explicitly account for such data heterogeneity. This can be justified by the homogeneity of data involved in typical machine learning and deep learning applications (e.g., a dataset of images or sentences). In contrast, hydrologic applications often encounter variables with different physical meaning, data representative at various scales (e.g., point-based ground stations, satellite imagery at different resolutions and sampling frequency), and noisy observations. In addition, measurements may contain bias and complex error structure that violate the commonly used white noise assumption. When these data are used as inputs and training targets, the data heterogeneity will likely affect the learning outcome. One way to account for heterogenous errors associated with training targets is to weigh the loss measured at each target inversely proportional to its uncertainty (Kendall et al., 2018) similarly as in weighted least squares regression (Tasker, 1980). However, methods to handle general input data uncertainty are still lacking.

Appropriately representing and propagating uncertainty is crucial for the robustness of predictions provided by the machine learning models particularly when they are trained with limited data and/or used under nonstationary conditions. Except for a few algorithms (e.g., GPR, Lasso), there has been a lack of theory for uncertainty quantification of conventional machine learning and deep learning models (Abdar et al., 2020). Some studies used ad hoc methods as a post-processing analysis to obtain prediction intervals (e.g., Solomatine et al., 2009; Xu et al., 2015). Ensemble learning methods (e.g., random forest) can produce uncertainty estimates by summarizing output from each ensemble member (Meinshausen, 2006; Tyralis et al., 2019). Ensemble methods have recently been applied to deep networks but tend to be computationally expensive (Osband et al., 2016; Pearce et al., 2018). In contrast to the frequentist approach based on ensembles, Bayesian neural networks reformulate the training problem as inferring the posterior distribution of weights (Heckerman, 2008; Ghahramani, 2015). However, exact Bayesian inference is computationally prohibitive for deep networks. Therefore, the posteriors are usually approximated using various methods such as Monte Carlo dropout at test time (Gal and Ghahramani, 2016) and variational autoencoders (Section 2.3.4). Nevertheless, the above methods only account for uncertainties in the network weights and cannot tackle data uncertainties.

Despite the reported successes, most of the studies reviewed in Section 3 are isolated applications of machine learning towards a specific problem. Often, deep learning architectures that have been tested and proven successful within the deep learning community need some tailoring before they can be applied to hydrologic problems. This is because a hydrologic application may not be directly mapped to a classical deep learning task for which these architectures have been established. For example, LSTMs have achieved great success for translating sentences from one language to another. A sentence differs from the time series of a hydrologic variable, and this difference affects the design of the deep learning architecture as well as data preparation practices. Often, identifying the appropriate architecture for a specific application requires substantial efforts involving trial-and-error, leading to a suboptimal choice. This difficulty partially counteracts the benefit deep learning offers in terms of avoiding feature engineering required by conventional machine learning methods. Bridging this disciplinary gap calls for formulation of hydrologic problems as “standard” machine learning tasks furnished with catered benchmark datasets.

5. CONCLUDING REMARKS

The recently revived interest within the hydrology community in machine learning in general and deep learning in particular is likely to continue given the hydrologic data deluge. The enormous amount of data poses challenges to traditional knowledge-driven reasoning and provides exciting opportunities for machine learning-based data-driven reasoning. In this overview, we attempted to provide a comprehensive, although far from complete, discussion of recent success stories of applying machine learning as a stand-alone model or complementary to process-based modeling efforts. Several primary challenges are identified in using machine learning for prediction under nonstationary conditions, developing interpretable machine learning models, ensuring physical consistency, training with limited sample size, and characterizing and propagating uncertainty. Meanwhile, there is emerging research that aims at integrating physical knowledge with machine learning to address some of the above challenges.

We argue that there is a need to develop formulations of representative hydrologic problems with quality-controlled benchmark datasets. These formulations can be related to one or more standard machine learning tasks that have been extensively studied, so that the advances in the machine learning and other fields can be leveraged to identify the best strategy to tackle the hydrologic problem. For example, forecasting of a hydrologic variable may be formulated as the problem of estimating the expected value (deterministic) or probability density function (probabilistic) of the variable of the next k time steps conditioned on historical measurements of itself and explanatory variables. Depending on how the variables are resolved spatially, each variable can be gridded or time series data. Such formulations will facilitate development of general-purpose architectures suitable for representative types of hydrologic applications as well as identifying similar problem formulations from other fields of geosciences. Data from isolated applications that fall within the same problem formulation can be compiled and quality controlled to create benchmark datasets that are much larger than data used in a single application. The benchmark datasets will serve as a venue for assessment and intercomparison of various machine learning models in terms of prediction capability, physical feasibility, and interpretability. Achieving this requires collective efforts within the hydrology community as well as interdisciplinary collaboration with the machine learning and geosciences communities.

Data Availability Statement

Data sharing is not applicable to this article as no new data were created or analyzed in this study.

Funding Information

T. Xu was supported by NOAA COM Grant NA20OAR4310341 and NSF Grant OAC-1931297 as well as funding provided by the School of Sustainable Engineering and the Built Environment, Ira A. Fulton Schools of Engineering, Arizona State University.

Acknowledgments

The authors thank Dr. Ruijie Zeng (Arizona State University) for comments on an earlier version of this manuscript and Qianqiu Longyang and Ruoyao Ou for their contributions to the visualizations. The authors claim no conflict of interest.

References

Abebe, A., and R. Price. (2003). Managing uncertainty in hydrological models using complementary models, *Hydrol. Sci. J., 48*(5), 679–692.

Aboutalebi, M., Allen, L. N., Torres-Rua, A. F., McKee, M., & Coopmans, C. (2019). Estimation of soil moisture at different soil levels using machine learning techniques and unmanned aerial vehicle (UAV) multispectral imagery. In *Autonomous Air and Ground Sensing Systems for Agricultural Optimization and Phenotyping IV* (Vol. 11008, p. 110080S). International Society for Optics and Photonics.

Abdar, M., Pourpanah, F., Hussain, S., Rezazadegan, D., Liu, L., Ghavamzadeh, M., ... & Nahavandi, S. (2020). A review of uncertainty quantification in deep learning: Techniques, applications and challenges. arXiv preprint arXiv:2011.06225.

Adnan, R. M., Liang, Z., Heddam, S., Zounemat-Kermani, M., Kisi, O., & Li, B. (2020). Least square support vector machine and multivariate adaptive regression splines for streamflow prediction in mountainous basin using hydro-meteorological data as inputs. *Journal of Hydrology*, *586*, 124371.

Ahmad, S., Kalra, A., & Stephen, H. (2010). Estimating soil moisture using remote sensing data: A machine learning approach. Advances in Water Resources, 33(1), 69–80. https://doi.org/10.1016/J.ADVWATRES.2009.10.008

Akbari Asanjan, A., Yang, T., Hsu, K., Sorooshian, S., Lin, J., & Peng, Q. (2018). Short-term precipitation forecast based on the PERSIANN system and LSTM recurrent neural networks. *Journal of Geophysical Research: Atmospheres*, *123*(22), 12-543.

Anda, A., Simon, B., Soós, G., Menyhárt, L., da Silva, J. A. T., & Kucserka, T. (2018). Extending Class A pan evaporation for a shallow lake to simulate the impact of littoral sediment and submerged macrophytes: a case study for Keszthely Bay (Lake Balaton, Hungary). *Agricultural and forest meteorology*, 250, 277-289.

Andugula, P., Durbha, S. S., Lokhande, A., & Suradhaniwar, S. (2017). Gaussian process based spatial modeling of soil moisture for dense soil moisture sensing network. In *2017 6th International Conference on Agro-Geoinformatics* (pp. 1-5). IEEE.

Asefa, T., Kemblowski, M., McKee, M., & Khalil, A. (2006). Multi-time scale stream flow predictions: The support vector machines approach. Journal of Hydrology, 318(1–4), 7–16. https://doi.org/10.1016/J.JHYDROL.2005.06.001

Asefa, T., Kemblowski, M., Urroz, G., McKee, M., 2005. Support vector machines (SVMs) for monitoring network design. Ground Water 43 (3), 413–422.

Asher, M. J., Croke, B. F., Jakeman, A. J., & Peeters, L. J. (2015). A review of surrogate models and their application to groundwater modeling. *Water Resources Research*, *51*(8), 5957-5973.

Ashouri, H., Hsu, K. L., Sorooshian, S., Braithwaite, D. K., Knapp, K. R., Cecil, L. D., ... & Prat, O. P. (2015). PERSIANN-CDR: Daily precipitation climate data record from multisatellite observations for hydrological and climate studies. *Bulletin of the American Meteorological Society*, *96*(1), 69-83.

Ayzel, G., & Heistermann, M. (2021). The effect of calibration data length on the performance of a conceptual hydrological model versus LSTM and GRU: A case study for six basins from the CAMELS dataset. *Computers & Geosciences*, 104708.

Bair, E. H., Abreu Calfa, A., Rittger, K., & Dozier, J. (2018). Using machine learning for real-time estimates of snow water equivalent in the watersheds of Afghanistan. *The Cryosphere*, *12*(5), 1579-1594.

Bardsley, W. E., Vetrova, V., & Liu, S. (2015). Toward creating simpler hydrological models: A LASSO subset selection approach. *Environmental Modelling & Software*, 72, 33-43.

Beck, H. E., van Dijk, A. I., De Roo, A., Miralles, D. G., McVicar, T. R., Schellekens, J., & Bruijnzeel, L. A. (2016). Global-scale regionalization of hydrologic model parameters. *Water Resources Research, 52*(5), 3599-3622.

Bengio, Y., Courville, A., & Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, *35*(8), 1798-1828.

Bengio, Y., Lamblin, P., Popovici, D., & Larochelle, H. (2007). Greedy layer-wise training of deep networks. In *Advances in neural information processing systems* (pp. 153-160).

Bengio, Y., Simard, P., & Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, *5*(2), 157-166.

Blanchet, F. G., Legendre, P., & Borcard, D. (2008). Forward selection of explanatory variables. *Ecology*, *89*(9), 2623-2632.

Boscarello, L., Ravazzani, G., Cislaghi, A., & Mancini, M. (2016). Regionalization of flow-duration curves through catchment classification with streamflow signatures and physiographic–climate indices. *Journal of Hydrologic Engineering*, *21*(3), 05015027.

Bottou, L. (2010). Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010* (pp. 177-186). Physica-Verlag HD.

Boucher, M.-A., Quilty, J., & Adamowski, J. (2020). Data assimilation for streamflow forecasting using extreme learning machines and multilayer perceptrons. *Water Resources Research*, *56*(6), e2019WR026226.

Boughton, W. C. (2007). Effect of data length on rainfall–runoff modeling. *Environmental Modeling & Software*, *22*(3), 406-413.

Breiman, Leo, 2001. Random forests. Mach. Learn. 45 (1), 5–32.

Brooks, W., Corsi, S., Fienen, M., & Carvin, R. (2016). Predicting recreational water quality advisories: A comparison of statistical methods. *Environmental Modeling & Software*, *76*, 81-94.

Broxton, P. D., Van Leeuwen, W. J., & Biederman, J. A. (2019). Improving snow water equivalent maps with machine learning of snow survey and lidar measurements. *Water Resources Research*, *55*(5), 3739-3757.

Brunton, S. L., Noack, B. R., & Koumoutsakos, P. (2020). Machine learning for fluid mechanics. *Annual Review of Fluid Mechanics*, 52, 477-508.

Buch, A. M., Mazumdar, H. S., & Pandey, P. C. (1993). Application of artificial neural networks in hydrological modeling: a case study of runoff simulation of a Himalayan glacier basin. In *Proceedings of 1993 International Conference on Neural Networks (IJCNN-93-Nagoya, Japan)* (Vol. 1, pp. 971-974). IEEE.

Burnham, K. P., & Anderson, D. R. (2004). Multimodel inference: understanding AIC and BIC in model selection. *Sociological methods & research*, *33*(2), 261-304.

Cai, X., Zeng, R., Kang, W. H., Song, J., & Valocchi, A. J. (2015). Strategic planning for drought mitigation under climate change. *Journal of Water Resources Planning and Management*, *141*(9), 04015004.

Chaney, N. W., Wood, E. F., McBratney, A. B., Hempel, J. W., Nauman, T. W., Brungard, C. W., & Odgers, N. P. (2016). POLARIS: A 30-meter probabilistic soil series map of the contiguous United States. *Geoderma*, *274*, 54-67.

Chaudhari, S., Mithal, V., Polatkan, G., & Ramanath, R. (2020). An Attentive Survey of Attention Models. *J. ACM*, 37, 4 (111).

Chen, H., Chandrasekar, V., Tan, H., & Cifelli, R. (2019). Rainfall estimation from ground radar and TRMM precipitation radar using hybrid deep neural networks. *Geophysical Research Letters*, *46*(17-18), 10669-10678.

Chen, T., & Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining* (pp. 785-794).

Chen, W., Tsangaratos, P., Ilia, I., Duan, Z., & Chen, X. (2019). Groundwater spring potential mapping using population-based evolutionary algorithms and data mining methods. *Science of The Total Environment*, *684*, 31-49.

Ching, T., Himmelstein, D. S., Beaulieu-Jones, B. K., Kalinin, A. A., Do, B. T., Way, G. P., ... & Greene, C. S. (2018). Opportunities and obstacles for deep learning in biology and medicine. *Journal of The Royal Society Interface*, *15*(141), 20170387.

Cho, E., Jacobs, J. M., Jia, X., & Kraatz, S. (2019). Identifying Subsurface Drainage using Satellite Big Data and Machine Learning via Google Earth Engine. *Water Resources Research*, *55*(10), 8028-8045.

Ciregan, D., Meier, U., & Schmidhuber, J. (2012). Multi-column deep neural networks for image classification. In *2012 IEEE conference on computer vision and pattern recognition* (pp. 3642-3649). IEEE.

Coates, A., Ng, A., & Lee, H. (2011). An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics* (pp. 215-223).

Creswell, A., White, T., Dumoulin, V., Arulkumaran, K., Sengupta, B., & Bharath, A. A. (2018). Generative adversarial networks: An overview. *IEEE Signal Processing Magazine*, *35*(1), 53-65.

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, *2*(4), 303-314.

de Bezenac, E., Pajot, A., & Gallinari, P. (2019). Deep learning for physical processes: Incorporating prior scientific knowledge. *Journal of Statistical Mechanics: Theory and Experiment*, *2019*(12), 124009.

de Oliveira, J. V., & Pedrycz, W. (Eds.). (2007). *Advances in fuzzy clustering and its applications*. John Wiley & Sons.

Deines, J. M., Kendall, A. D., & Hyndman, D. W. (2017). Annual irrigation dynamics in the US Northern High Plains derived from Landsat satellite data. *Geophysical Research Letters*, *44*(18), 9350-9360.

Demissie, Y. K., Valocchi, A. J., Minsker, B. S., & Bailey, B. A. (2009). Integrating a calibrated groundwater flow model with error-correcting data-driven models to improve predictions. *Journal of hydrology*, *364*(3-4), 257-271.

Demissie, Y., Valocchi, A., Cai, X., Brozovic, N., Senay, G., & Gebremichael, M. (2015). Parameter estimation for groundwater models under uncertain irrigation data. *Groundwater*, *53*(4), 614-625.

Ding, Y., Wang, L., Fan, D., & Gong, B. (2018, March). A semi-supervised two-stage approach to learning from noisy labels. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)* (pp. 1215-1224). IEEE.

Ding, Y., Zhu, Y., Wu, Y., Jun, F., & Cheng, Z. (2019). Spatio-Temporal Attention LSTM Model for Flood Forecasting. In *2019 International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)* (pp. 458-465). IEEE.

Du, S. S., Wang, Y., Zhai, X., Balakrishnan, S., Salakhutdinov, R., & Singh, A. (2018). How many samples are needed to estimate a convolutional or recurrent neural network?. *arXiv preprint arXiv:1805.07883*.

Elkahky, A. M., Song, Y., & He, X. (2015). A multi-view deep learning approach for cross domain user modeling in recommendation systems. In *Proceedings of the 24th International Conference on World Wide Web* (pp. 278-288).

Evin, G., Thyer, M., Kavetski, D., McInerney, D., & Kuczera, G. (2014). Comparison of joint versus postprocessor approaches for hydrological uncertainty estimation accounting for error autocorrelation and heteroscedasticity. *Water Resources Research*, *50*(3), 2350-2375. doi:10.1002/ 2013WR014185

Fang, K., Shen, C., Kifer, D., & Yang, X. (2017). Prolongation of SMAP to spatio-temporally seamless coverage of continental US using a deep learning neural network. Geophysical Research Letters, 44, 11,030–11,039. https://doi.org/10.1002/2017GL075619

Fang, K., & Shen, C. (2020). Near-real-time forecast of satellite-based soil moisture using long short-term memory with an adaptive data integration kernel. *Journal of Hydrometeorology*, *21*(3), 399-413.

Fernández-Delgado, M., Cernadas, E., Barro, S., & Amorim, D. (2014). Do we need hundreds of classifiers to solve real world classification problems?. *The journal of machine learning research*, *15*(1), 3133-3181.

Fernández-Delgado, M., Sirsat, M. S., Cernadas, E., Alawadi, S., Barro, S., & Febrero-Bande, M. (2019). An extensive experimental survey of regression methods. *Neural Networks*, *111*, 11-34.

Frame, J., Nearing, G., Kratzert, F., & Rahman, M. (2020). Post processing the US National Water Model with a Long Short-Term Memory network. https://doi.org/10.31223/osf.io/4xhac

Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of statistics*, 1189-1232.

Gal, Y., & Ghahramani, Z. (2016). Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning* (pp. 1050-1059).

Gao, Q., Zribi, M., Escorihuela, M. J., Baghdadi, N., & Segui, P. Q. (2018). Irrigation mapping using Sentinel-1 time series at field scale. *Remote Sensing*, *10*(9), 1495.

George, E. I. (2000). The variable selection problem. *Journal of the American Statistical Association*, *95*(452), 1304-1308.

Géron, A. (2019). *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. O'Reilly Media.

Ghahramani, Z. (2015). Probabilistic machine learning and artificial intelligence. *Nature*, *521*(7553), 452-459.

Ghose, D., Das, U., & Roy, P. (2018). Modeling response of runoff and evapotranspiration for predicting water table depth in arid region using dynamic recurrent neural network. *Groundwater for Sustainable Development*, *6*, 263-269.

Gilpin, L. H., Bau, D., Yuan, B. Z., Bajwa, A., Specter, M., & Kagal, L. (2018). Explaining explanations: An overview of interpretability of machine learning. In *2018 IEEE 5th International Conference on data science and advanced analytics (DSAA)* (pp. 80-89). IEEE.

Glorot, X., & Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics* (pp. 249-256).

Glorot, X., Bordes, A. & Bengio. Y. (2011). Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics* (pp. 315-323).

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2014). Generative adversarial nets. In *Advances in neural information processing systems* (pp. 2672-2680).

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT press.

Goodwell, A. E., & Kumar, P. (2017). Temporal information partitioning: Characterizing synergy, uniqueness, and redundancy in interacting environmental variables. *Water Resources Research*, *53*(7), 5920-5942.

Gorelick, N., Hancher, M., Dixon, M., Ilyushchenko, S., Thau, D., & Moore, R. (2017). Google Earth Engine: Planetary-scale geospatial analysis for everyone. *Remote sensing of Environment*, *202*, 18-27.

Graves, A. (2012). Long short-term memory. In *Supervised sequence labelling with recurrent neural networks* (pp. 37-45). Springer, Berlin, Heidelberg.

Greff, K., Srivastava, R. K., Koutník, J., Steunebrink, B. R., & Schmidhuber, J. (2017). LSTM: A search space odyssey. *IEEE transactions on neural networks and learning systems*, *28*(10), 2222-2232.

Gupta, V. K., & Sorooshian, S. (1985). The relationship between data and the precision of parameter estimates of hydrologic models. *Journal of Hydrology*, *81*(1-2), 57-77.

Gupta, H. V., Sorooshian, S., & Yapo, P. O. (1998). Toward improved calibration of hydrologic models: Multiple and noncommensurable measures of information. *Water Resources Research*, *34*(4), 751-763.

Gusyev, M. A., Haitjema, H. M., Carlson, C. P., & Gonzalez, M. A. (2013). Use of nested flow models and interpolation techniques for science-based management of the Sheyenne National Grassland, North Dakota, USA. *Groundwater*, *51*(3), 414-420.

Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of machine learning research*, *3*(Mar), 1157-1182.

Guzman, S. M., Paz, J. O., Tagert, M. L. M., & Mercer, A. E. (2019). Evaluation of seasonally classified inputs for the prediction of daily groundwater levels: NARX networks vs support vector machines. *Environmental Modeling & Assessment*, *24*(2), 223-234.

Hartigan, J. A., & Wong, M. A. (1979). Algorithm AS 136: A k-means clustering algorithm. *Journal of the royal statistical society. series c (applied statistics)*, *28*(1), 100-108.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning: data mining, inference, and prediction*, 2nd Ed. Springer Science & Business Media.

Heckerman, D. (2008). A tutorial on learning with Bayesian networks. *Innovations in Bayesian networks*, edited by Holmes, D. E. and Jain, L. C. Springer.

Hino, M., Benami, E., & Brooks, N. (2018). Machine learning for environmental monitoring. *Nature Sustainability*, *1*(10), 583-588.

Hipsey, M. R., Hamilton, D. P., Hanson, P. C., Carey, C. C., Coletti, J. Z., Read, J. S., ... & Brookes, J. D. (2015). Predicting the resilience and recovery of aquatic systems: A framework for model evolution within environmental observatories. *Water Resources Research*, *51*(9), 7023-7043.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural computation, 9(8), 1735-1780.

Hochreiter, S. (1998). The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, *6*(02), 107-116.

Hofmann, T., Schölkopf, B., & Smola, A. J. (2008). Kernel methods in machine learning. *The annals of statistics*, 1171-1220.

Hosmer Jr, D. W., Lemeshow, S., & Sturdivant, R. X. (2013). *Applied logistic regression* (Vol. 398). John Wiley & Sons.

Hsu, K. L., Gupta, H. V., & Sorooshian, S. (1995). Artificial neural network modeling of the rainfall-runoff process. *Water resources research*, *31*(10), 2517-2530.

Hu, R., Fang, F., Pain, C. C., & Navon, I. M. (2019). Rapid spatio-temporal flood prediction and uncertainty quantification using a deep learning method. *Journal of Hydrology*, *575*, 911-920.

Hu, Y., Quinn, C. J., Cai, X., & Garfinkle, N. W. (2017). Combining human and machine intelligence to derive agents' behavioral rules for groundwater irrigation. *Advances in water resources*, *109*, 29-40.

Hutter, F., Kotthoff, L., & Vanschoren, J. (2019). *Automated machine learning: methods, systems, challenges* (p. 219). Springer Nature.

Irani, J., Pise, N., & Phatak, M. (2016). Clustering techniques and the similarity measures used in clustering: a survey. *International journal of computer applications*, *134*(7), 9-14.

Izenman, A. J. (2013). Linear discriminant analysis. In *Modern multivariate statistical techniques* (pp. 237-280). Springer, New York, NY.

Jia, X., Willard, J., Karpatne, A., Read, J., Zwart, J., Steinbach, M., & Kumar, V. (2019). Physics guided RNNs for modeling dynamical systems: A case study in simulating lake temperature profiles. In *Proceedings of the 2019 SIAM International Conference on Data Mining* (pp. 558-566). Society for Industrial and Applied Mathematics.

Jain, A. K. (2010). Data clustering: 50 years beyond K-means. *Pattern recognition letters*, *31*(8), 651-666.

Jiang, S., Zheng, Y., & Solomatine, D. (2020). Improving AI system awareness of geoscience knowledge: Symbiotic integration of physical approaches and deep learning. *Geophysical Research Letters*, *47*(13), e2020GL088229.

Kamrava, S., Tahmasebi, P., & Sahimi, M. (2020). Linking morphology of porous media to their macroscopic permeability by deep learning. *Transport in Porous Media*, *131*(2), 427-448.

Kang, K. W., Park, C. Y., & Kim, J. H. (1993). Neural network and its application to rainfall-runoff forecasting. *Korean Journal of Hydrosciences*, *4*, 1-9.

Kaplan, A., & Haenlein, M. (2019). Siri, Siri, in my hand: Who's the fairest in the land? On the interpretations, illustrations, and implications of artificial intelligence. *Business Horizons*, *62*(1), 15-25.

Karpatne, A., Ebert-Uphoff, I., Ravela, S., Babaie, H. A., & Kumar, V. (2019). Machine learning for the geosciences: Challenges and opportunities. *IEEE Transactions on Knowledge and Data Engineering*, *31*(8), 1544-1554.

Ke, Y., Im, J., Park, S., & Gong, H. (2016). Downscaling of MODIS One kilometer evapotranspiration using Landsat-8 data and machine learning approaches. *Remote Sensing*, *8*(3), 215.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., ... & Liu, T. Y. (2017). Lightgbm: A highly efficient gradient boosting decision tree. In *Advances in neural information processing systems* (pp. 3146-3154).

Kendall, A., Gal, Y., & Cipolla, R. (2018). Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 7482-7491).

Kennedy, M. C., & O'Hagan, A. (2001). Bayesian calibration of computer models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *63*(3), 425-464.

Khan, S., & Yairi, T. (2018). A review on the application of deep learning in system health management. *Mechanical Systems and Signal Processing*, *107*, 241-265.

Kim, S., Kim, H., Lee, J., Yoon, S., Kahou, S. E., Kashinath, K., & Prabhat, M. (2019, January). Deep-hurricane-tracker: Tracking and forecasting extreme climate events. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)* (pp. 1761-1769). IEEE.

Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*.

Kingma, D. P., Mohamed, S., Rezende, D. J., & Welling, M. (2014). Semi-supervised learning with deep generative models. In *Advances in neural information processing systems* (pp. 3581-3589).

Kingma, D. P., & Welling, M. (2014). Auto-encoding variational bayes. In *International Conference on Learning Representations (ICLR)*.

Kleiber, W., Katz, R. W., and Rajagopalan, B. (2012), Daily spatiotemporal precipitation simulation using latent and transformed Gaussian processes, *Water Resour. Res.*, 48, W01523, doi:10.1029/2011WR011105.

Kordestani, M. D., Naghibi, S. A., Hashemi, H., Ahmadi, K., Kalantar, B., & Pradhan, B. (2019). Groundwater potential mapping using a novel data-mining ensemble model. *Hydrogeol. J*, *27*(1), 211-224. https://doi.org/10.1007/s10040-018-1848-5.

Kratzert, F., Klotz, D., Brenner, C., Schulz, K., & Herrnegger, M. (2018). Rainfall–runoff modeling using long short-term memory (LSTM) networks. *Hydrology and Earth System Sciences*, *22*(11), 6005-6022.

Kratzert, F., Herrnegger, M., Klotz, D., Hochreiter, S., & Klambauer, G. (2019a). NeuralHydrology–Interpreting LSTMs in Hydrology. In *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning* (pp. 347-362). Springer, Cham.

Kratzert, F., Klotz, D., Shalev, G., Klambauer, G., Hochreiter, S., & Nearing, G. (2019b). Towards learning universal, regional, and local hydrological behaviors via machine learning applied to large-sample datasets. *Hydrology & Earth System Sciences*, *23*(12).

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097-1105).

Kumar, P. (2015). Hydrocomplexity: Addressing water security and emergent environmental risks. *Water Resources Research*, *51*(7), 5827-5838.

Laloy, E., Hérault, R., Lee, J., Jacques, D., & Linde, N. (2017). Inversion using a new low-dimensional representation of complex binary geological media based on a deep neural network. *Advances in Water Resources*, *110*, 387-405.

Laloy, E., Hérault, R., Jacques, D., & Linde, N. (2018). Training-image based geostatistical inversion using a spatial generative adversarial neural network. *Water Resources Research*, *54*(1), 381-406.

Laloy, E., & Jacques, D. (2019). Emulation of CPU-demanding reactive transport models: a comparison of Gaussian processes, polynomial chaos expansion, and deep neural networks. *Computational Geosciences*, *23*(5), 1193-1215.

Laloy, E., Linde, N., Ruffino, C., Hérault, R., Gasso, G., & Jacques, D. (2019). Gradient-based deterministic inversion of geophysical data with generative adversarial networks: Is it feasible?. *Computers & Geosciences*, *133*, 104333.

LeCun, Y., Boser, B. E., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W. E., & Jackel, L. D. (1990). Handwritten digit recognition with a back-propagation network. In *Advances in neural information processing systems* (pp. 396-404).

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition, *Proceedings of the IEEE, 86*(11), 2278–2324.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *nature*, *521*(7553), 436-444.

Lee, C.suk, Sohn, E., Park, J.D., Jang, J.-.D., 2019. Estimation of soil moisture using deep learning based on satellite data: a case study of South Korea. GISci. Remote Sens. 56, 43–67. https://doi.org/10.1080/15481603.2018.1489943.

Li, M., Zhang, T., Chen, Y., & Smola, A. J. (2014). Efficient mini-batch training for stochastic optimization. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 661-670).

Liang, F., Mao, K., Liao, M., Mukherjee, S., & West, M. (2007). Nonparametric Bayesian kernel models. *Department of Statistical Science, Duke University, Discussion Paper*, 07-10.

Liang, F., Paulo, R., Molina, G., Clyde, M. A., & Berger, J. O. (2008). Mixtures of g priors for Bayesian variable selection. *Journal of the American Statistical Association*, *103*(481), 410-423.

Liu, Y., & Gupta, H. V. (2007). Uncertainty in hydrologic modeling: Toward an integrated data assimilation framework. *Water resources research*, *43*(7).

Liu, H., Ong, Y. S., Shen, X., & Cai, J. (2020). When Gaussian process meets big data: A review of scalable GPs. *IEEE transactions on neural networks and learning systems*, *31*(11), 4405-4423.

Liu, Y., Racah, E., Correa, J., Khosrowshahi, A., Lavers, D., Kunkel, K., ... & Collins, W. (2016). Application of deep convolutional neural networks for detecting extreme weather in climate datasets. *arXiv preprint arXiv:1605.01156*.

Liu, F., Xu, F., & Yang, S. (2017). A flood forecasting model based on deep learning algorithm via integrating stacked autoencoders with BP neural network. In *2017 IEEE third International conference on multimedia big data (BigMM)* (pp. 58-61). IEEE.

Liu, S., Zhong, Z., Takbiri-Borujeni, A., Kazemi, M., Fu, Q., & Yang, Y. (2019). A case study on homogeneous and heterogeneous reservoir porous media reconstruction by using generative adversarial networks. *Energy Procedia*, *158*, 6164-6169.

Lv, N., Liang, X., Chen, C., Zhou, Y., Li, J., Wei, H., & Wang, H. (2020). A Long Short-Term Memory Cyclic model With Mutual Information For Hydrology Forecasting: A Case Study in the Xixian Basin. *Advances in Water Resources*, 103622.

Ma, Y., Montzka, C., Bayat, B., & Kollet, S. (2020). Using Long Short-Term Memory networks to connect water table depth anomalies to precipitation anomalies over Europe. *Hydrology and Earth System Sciences Discussions*, 1-30.

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability* (Vol. 1, No. 14, pp. 281-297).

Mater, A. C., & Coote, M. L. (2019). Deep learning in chemistry. *Journal of chemical information and modeling*, *59*(6), 2545-2559.

Meempatta, L., Webb, A. J., Horne, A. C., Keogh, L. A., Loch, A., & Stewardson, M. J. (2019). Reviewing the decision-making behavior of irrigators. *Wiley Interdisciplinary Reviews: Water*, *6*(5), e1366.

Meinshausen, N. (2006). Quantile regression forests. *Journal of Machine Learning Research*, *7*(Jun), 983-999.

Mishkin, D., & Matas, J. (2015). All you need is a good init. *arXiv preprint arXiv:1511.06422*.

Mitchell, M. (1997). Machine learning. *Burr Ridge, IL: McGraw Hill, 45*(37), 870-877.

Mo, S., Zabaras, N., Shi, X., Wu, J. (2019a). 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., Zhu, Y., Zabaras, N. J., Shi, X., & Wu, J. (2019b). Deep convolutional encoder-decoder networks for uncertainty quantification of dynamic multiphase flow in heterogeneous media. *Water Resources Research, 55*(1), 703–728. https://doi.org/10.1029/2018WR023528.

Moghaddam, D. D., Rahmati, O., Panahi, M., Tiefenbacher, J., Darabi, H., Haghizadeh, A., ... & Bui, D. T. (2020). The effect of sample size on different machine learning models for groundwater potential mapping in mountain bedrock aquifers. *Catena*, *187*, 104421.

Moghaddam, M. A., Ferre, P. A., Chen, X., Chen, K., Song, X., & Hammond, G. E. (2020). Applying Simple Machine Learning Tools to Infer Streambed Flux from Subsurface Pressure and Temperature Observations.

Murtagh, F., & Legendre, P. (2014). Ward's hierarchical agglomerative clustering method: which algorithms implement Ward's criterion?. *Journal of classification*, *31*(3), 274-295.

Naghibi, S. A., Ahmadi, K., & Daneshi, A. (2017). Application of support vector machine, random forest, and genetic algorithm optimized random forest models in groundwater potential mapping. *Water Resources Management*, *31*(9), 2761-2775.

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), 1-21.

Nearing, G., Sampson, A. K., Kratzert, F., & Frame, J. (2020). Post-processing a Conceptual Rainfall-runoff Model with an LSTM. https://doi.org/10.31223/osf.io/53te4.

Nolan, B. T., Fienen, M. N., & Lorenz, D. L. (2015). A statistical learning framework for groundwater nitrate models of the Central Valley, California, USA. *Journal of Hydrology*, *531*, 902-911.

Osband, I., Blundell, C., Pritzel, A. and Van Roy, B. (2016) Deep exploration via bootstrapped DQN. In Proceeding NeurIPS 2016.

Pan, B., Hsu, K., AghaKouchak, A., & Sorooshian, S. (2019). Improving precipitation estimation using convolutional neural network. *Water Resources Research, 55*, 2301–2321. https://doi.org/10.1029/2018WR024090

Pande, S., & Sivapalan, M. (2017). Progress in socio-hydrology: A meta-analysis of challenges and opportunities. *Wiley Interdisciplinary Reviews: Water*, *4*(4), e1193.

Pearce, T., Brintrup, A., Zaki, M., & Neely, A. (2018). High-quality prediction intervals for deep learning: A distribution-free, ensembled approach. In Proceeding of International Conference on Machine Learning (pp. 4075-4084).

Phan, N., Dou, D., Wang, H., Kil, D., & Piniewski, B. (2017). Ontology-based deep learning for human behavior prediction with explanations in health social networks. *Information sciences*, *384*, 298-313.

Pianosi, F., & Raso, L. (2012). Dynamic modeling of predictive uncertainty by regression on absolute errors. *Water Resources Research*, *48*(3). W03516, doi:10.1029/2011WR010603.

Prechelt, L. (1998). Automatic early stopping using cross validation: quantifying the criteria. *Neural Networks*, *11*(4), 761-767.

Radovic, A., Williams, M., Rousseau, D., Kagan, M., Bonacorsi, D., Himmel, A., ... & Wongjirad, T. (2018). Machine learning at the energy and intensity frontiers of particle physics. *Nature*, *560*(7716), 41-48.

Rasmussen, C. E., and C. K. I. Williams (2006), *Gaussian Processes for Machine Learning*, MIT Press, Cambridge, Mass.

Rasouli, K., Hsieh, W. W., & Cannon, A. J. (2012). Daily streamflow forecasting by machine learning methods with weather and climate inputs. *Journal of Hydrology*, *414*, 284-293.

Racah, E., Beckham, C., Maharaj, T., Kahou, S. E., Prabhat, M., & Pal, C. (2017). ExtremeWeather: A large-scale climate dataset for semi-supervised detection, localization, and understanding of extreme weather events. In *Advances in Neural Information Processing Systems* (pp. 3402-3413).

Razavi, S., & Tolson, B. A. (2013). An efficient framework for hydrologic model calibration on long data periods. *Water Resources Research*, *49*(12), 8418-8431.

Reichstein, M., Camps-Valls, G., Stevens, B., Jung, M., Denzler, J., & Carvalhais, N. (2019). Deep learning and process understanding for data-driven Earth system science. *Nature*, *566*(7743), 195-204.

Ren, W. W., Yang, T., Huang, C. S., Xu, C. Y., & Shao, Q. X. (2018). Improving monthly streamflow prediction in alpine regions: integrating HBV model with Bayesian neural network. *Stochastic Environmental Research and Risk Assessment*, *32*(12), 3381-3396.

Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*. *65*(6), 386-408. doi:10.1037/h0042519.

Rudin, C. (2019). Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, *1*(5), 206-215.

Rumelhart, D. E., Hinton, G. E. & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature, 323*(6088), 533-536.

Sahoo, S., Russo, T. A., Elliott, J., & Foster, I. (2017). Machine learning algorithms for modeling groundwater level changes in agricultural regions of the US. *Water Resources Research*, *53*(5), 3878-3895.

Samek, W., & Müller, K. R. (2019). Towards explainable artificial intelligence. In *Explainable AI: interpreting, explaining and visualizing deep learning* (pp. 5-22). Springer, Cham.

Sawicz, K. A., Kelleher, C., Wagener, T., Troch, P., Sivapalan, M., & Carrillo, G. (2014). Characterizing hydrologic change through catchment classification. *Hydrology and Earth System Sciences*, *18*(1), 273.

Saxe, A. M., Koh, P. W., Chen, Z., Bhand, M., Suresh, B., & Ng, A. Y. (2011). On random weights and unsupervised feature learning. In *ICML* (Vol. 2, No. 3, p. 6).

Sengupta, S., Basak, S., Saikia, P., Paul, S., Tsalavoutis, V., Atiah, F., ... & Peters, A. (2020). A review of deep learning with special emphasis on architectures, applications and recent trends. *Knowledge-Based Systems*, *194*, 105596.

Settles, B. (2011). From theories to queries: Active learning in practice. In *Active Learning and Experimental Design workshop In conjunction with AISTATS 2010* (pp. 1-18).

Seyoum, W. M., Kwon, D., & Milewski, A. M. (2019). Downscaling GRACE TWSA data into high-resolution groundwater level anomaly using machine learning-based models in a glacial aquifer system. *Remote Sensing*, *11*(7), 824.

Shen, C. (2018). A transdisciplinary review of deep learning research and its relevance for water resources scientists. *Water Resources Research*, *54*(11), 8558-8593.

Shen, C., Laloy, E., Elshorbagy, A., Albert, A., Bales, J., Chang, F. J., ... & Fang, K. (2018). HESS Opinions: Incubating deep-learning-powered hydrologic science advances as a community. *Hydrology and Earth System Sciences (Online)*, *22*(11).

Shi, X., Chen, Z., Wang, H., Yeung, D. Y., Wong, W. K., & Woo, W. C. (2015). Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In *Advances in neural information processing systems* (pp. 802-810).

Shi, X., Gao, Z., Lausen, L., Wang, H., Yeung, D. Y., Wong, W. K., & Woo, W. C. (2017). Deep learning for precipitation nowcasting: A benchmark and a new model. In *Advances in neural information processing systems* (pp. 5617-5627).

Singh, S. K., & Bárdossy, A. (2012). Calibration of hydrological models on hydrologically unusual events. *Advances in Water Resources*, *38*, 81-91.

Smith, J., & Eli, R. N. (1995). Neural-network models of rainfall-runoff process. *Journal of water resources planning and management*, *121*(6), 499-508.

Smith, R. G., & Majumdar, S. (2020). Groundwater storage loss associated with land subsidence in Western United States mapped using machine learning. *Water Resources Research, 56*(7), e2019WR026621. https://doi.org/ 10.1029/2019WR026621

Sohangir, S., Wang, D., Pomeranets, A., & Khoshgoftaar, T. M. (2018). Big Data: Deep Learning for financial sentiment analysis. *Journal of Big Data*, *5*(1), 3.

Solomatine, D. P., & Shrestha, D. L. (2009). A novel method to estimate model uncertainty using machine learning techniques. *Water Resources Research*, *45*(12). W00B11, doi:10.1029/2008WR006839.

Sønderby, C. K., Raiko, T., Maaløe, L., Sønderby, S. K., & Winther, O. (2016). Ladder variational autoencoders. In *Advances in neural information processing systems* (pp. 3738-3746).

Sorooshian, S., Hsu, K. L., Gao, X., Gupta, H. V., Imam, B., & Braithwaite, D. (2000). Evaluation of PERSIANN system satellite-based estimates of tropical rainfall. *Bulletin of the American Meteorological Society*, *81*(9), 2035-2046.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, *15*(1), 1929-1958.

Sun, A. Y., Scanlon, B. R., Zhang, Z., Walling, D., Bhanja, S. N., Mukherjee, A., & Zhong, Z. (2019). Combining physically based modeling and deep learning for fusing GRACE satellite data: Can we learn from mismatch?. *Water Resources Research*, *55*(2), 1179-1195. https://doi.org/10.1029/2018WR023333

Sutskever, I., Martens, J., Dahl, G., & Hinton, G. (2013, February). On the importance of initialization and momentum in deep learning. In *International conference on machine learning* (pp. 1139-1147).

Tahmasebi, P., Kamrava, S., Bai, T., & Sahimi, M. (2020). Machine learning in geo-and environmental sciences: From small to large scale. *Advances in Water Resources*, 103619.

Tao, Y., Gao, X., Hsu, K., Sorooshian, S., & Ihler, A. (2016). A deep neural network modeling framework to reduce bias in satellite precipitation products. *Journal of Hydrometeorology*, *17*(3), 931-945.

Tartakovsky, A. M., Marrero, C. O., Perdikaris, P., Tartakovsky, G. D., & Barajas-Solano, D. (2020). Physics-informed deep neural networks for learning parameters and constitutive relationships in subsurface flow problems. *Water Resources Research, 56*(5), e2019WR026731. https://doi.org/10. 1029/2019WR026731

Tasker, G. D. (1980). Hydrologic regression with weighted least squares. *Water Resources Research*, 16(6), 1107-1113.

Tennant, C., Larsen, L., Bellugi, D., Moges, E., Zhang, L., & Ma, H. (2020). The utility of information flow in formulating discharge forecast models: a case study from an arid snow-dominated catchment. *Water Resources Research*, *56*(8), e2019WR024908.

Tennant, H., Neilson, B. T., Miller, M. P., & Xu, T. (2021). Ungaged inflow and loss patterns in urban and agricultural sub-reaches of the Logan River Observatory. *Hydrological Processes*, doi: 10.1002/hyp.14097.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*. *58*(1),267-288.

Torres, A. F., Walker, W. R., & McKee, M. (2011). Forecasting daily potential evapotranspiration using machine learning and limited climatic data. *Agricultural Water Management*, *98*(4), 553-562.

Toth, E. (2013). Catchment classification based on characterisation of streamflow and precipitation time series. *Hydrology & Earth System Sciences*, *17*(3).

Turing, A. (1950). Computing Machinery and Intelligence. *Mind*. *59*(236), 433–460. doi:10.1093/mind/LIX.236.433

Tyralis, H., Papacharalampous, G., Burnetas, A., & Langousis, A. (2019). Hydrological post-processing using stacked generalization of quantile regression algorithms: Large-scale application over CONUS. *Journal of Hydrology*, *577*, 123957.

Van den Oord, A., Dieleman, S., & Schrauwen, B. (2013). Deep content-based music recommendation. In *Advances in neural information processing systems* (pp. 2643-2651).

Vandal, T., Kodra, E., & Ganguly, A. R. (2019). Intercomparison of machine learning methods for statistical downscaling: the case of daily and extreme precipitation. *Theoretical and Applied Climatology*, *137*(1), 557-570.

Vapnik, V.N. (1995). *The Nature of Statistical Learning Theory*. New York: Springer.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems* (pp. 5998-6008).

Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., Manzagol, P. A., & Bottou, L. (2010). Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of machine learning research*, *11*(12).

Vinyals, O., Toshev, A., Bengio, S., & Erhan, D. (2015). Show and tell: A neural image caption generator. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3156-3164).

Wang, C., Duan, Q., Gong, W., Ye, A., Di, Z., & Miao, C. (2014). An evaluation of adaptive surrogate modeling based optimization with two benchmark problems. *Environmental Modeling & Software*, *60*, 167-179.

Wang, N., Zhang, D., Chang, H., & Li, H. (2020). Deep learning of subsurface flow via theory-guided neural network. *Journal of Hydrology*, *584*, 124700.

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.

Wu, H., Fang, W. Z., Kang, Q., Tao, W. Q., & Qiao, R. (2019). Predicting effective diffusivity of porous media from images by deep learning. *Scientific reports*, *9*(1), 1-12. https://doi.org/10.1038/s41598-019-56309-x.

Wu, J., Yin, X., & Xiao, H. (2018). Seeing permeability from images: fast prediction with convolutional neural networks. *Science bulletin*, *63*(18), 1215-1222. https://doi.org/10.1016/j.scib.2018.08.006/

Wunsch, A., Liesch, T., & Broda, S. (2018). Forecasting groundwater levels using nonlinear autoregressive networks with exogenous input (NARX). *Journal of Hydrology*, *567*, 743-758.

Xiang, Z., Yan, J., & Demir, I. (2020). A rainfall-runoff model with LSTM-based sequence-to-sequence learning. *Water resources research*, *56*(1), e2019WR025326.

Xu, T., & Valocchi, A. J. (2015). Data-driven methods to improve baseflow prediction of a regional groundwater model. *Computers & Geosciences*, 85, 124-136.

Xu, T., Valocchi, A. J., Ye, M., & Liang, F. (2017). Quantifying model structural error: Efficient Bayesian calibration of a regional groundwater flow model using surrogates and a data-driven error model. *Water Resources Research*, *53*(5), 4084-4105. doi:10.1002/ 2016WR019831.

Xu, T., Deines, J. M., Kendall, A. D., Basso, B., & Hyndman, D. W. (2019). Addressing challenges for mapping irrigated fields in subhumid temperate regions by integrating remote sensing and hydroclimatic data. *Remote Sensing*, *11*(3), 370.

Xu, T. R., Guo, Z., Liu, S., He, X., Meng, Y., Xu, Z., ... & Song, L. (2018). Evaluating different machine learning methods for upscaling evapotranspiration from flux towers to the regional scale. *Journal of Geophysical Research: Atmospheres*, *123*(16), 8674-8690.

Yang, J., Jakeman, A., Fang, G., & Chen, X. (2018). Uncertainty analysis of a semi-distributed hydrologic model based on a Gaussian Process emulator. *Environmental Modeling & Software*, *101*, 289-300.

Yapo, P. O., Gupta, H. V., & Sorooshian, S. (1996). Automatic calibration of conceptual rainfall-runoff models: sensitivity to calibration data. *Journal of Hydrology*, *181*(1-4), 23-48.

Yaseen, Z. M., El-Shafie, A., Jaafar, O., Afan, H. A., & Sayl, K. N. (2015). Artificial intelligence based models for stream-flow forecasting: 2000–2015. *Journal of Hydrology*, *530*, 829-844.

Yoon, H., Jun, S.-C., Hyun, Y., Bae, G.-O., & Lee, K.-K. (2011). A comparative study of artificial neural networks and support vector machines for predicting groundwater levels in a coastal aquifer. Journal of Hydrology, 396(1–2), 128–138. https://doi.org/10.1016/J. JHYDROL.2010.11.002Yuan, Q., Shen, H., Li, T., Li, Z., Li, S., Jiang, Y., ... & Zhang, L. (2020). Deep learning in environmental remote sensing: Achievements and challenges. *Remote Sensing of Environment*, *241*, 111716.

Zeng, R., Cai, X., Ringler, C., & Zhu, T. (2017). Hydropower versus irrigation—an analysis of global patterns. *Environmental Research Letters*, *12*(3), 034006.

Zhang, D., Zhang, W., Huang, W., Hong, Z., & Meng, L. (2017). Upscaling of surface soil moisture using a deep learning model with VIIRS RDR. *ISPRS International Journal of Geo-Information*, *6*(5), 130.

Zhang, J., Zhu, Y., Zhang, X., Ye, M., & Yang, J. (2018). Developing a Long Short-Term Memory (LSTM) based model for predicting water table depth in agricultural areas. *Journal of hydrology*, *561*, 918-929. https://doi.org/10.1016/j.jhydrol.2018.04.065.

Zhang, J., Zheng, Q., Chen, D., Wu, L., & Zeng, L. (2020). Surrogate-Based Bayesian Inverse Modeling of the Hydrological System: An Adaptive Approach Considering Surrogate Approximation Error. Water Resources Research, 56, e2019WR025721. https:// doi.org/10.1029/2019WR025721

Zhao, W. L., Gentine, P., Reichstein, M., Zhang, Y., Zhou, S., Wen, Y., ... & Qiu, G. Y. (2019). Physics-constrained machine learning of evapotranspiration. *Geophysical Research Letters*, *46*(24), 14496-14507.

Zhu, X., & Goldberg, A. B. (2009). Introduction to semi-supervised learning. *Synthesis lectures on artificial intelligence and machine learning*, *3*(1), 1-130.