Applying Simple Machine Learning Tools to Infer Streambed Flux from Subsurface Pressure and Temperature Observations

MA Moghaddam[1], Paul A. Ferre[1], Xingyuan Chen[2], Kewei Chen[3], Xuehang Song[4], and Glenn Edward Hammond[5]

[1]University of Arizona
[2]Pacific Northwest National Laboratory (DOE)
[3]Pacific Northwest National Laboratory
[4]PNNL
[5]Sandia National Laboratories

November 22, 2022

Abstract

We demonstrate the application of two simple machine learning tools - regression tree and gradient boosting analyses - to a hydrologic inference problem to address two objectives. The first goal was to infer the flux between a river and the subsurface based on high temporal resolution (5-minute) observations of subsurface pressure and temperature. The second goal was to identify an optimal set of observations to support these inferences. Specifically, we examine how many and what type of observations (pressure and/or temperature) were necessary and at what depths. Using synthetic observations and surface fluxes provided by a fully resolved three-dimensional flow and heat transport model, we found that both machine learning tools could identify the flux well using pressure and temperature measurements collected at three depths, even when considerable noise was added to the synthetic observations. Neither method could provide reasonable flux estimates given only noisy temperature data. A shallow, collocated temperature and pressure observations performed as well as the complete data set. The results show the promise of using machine learning tools to design hydrologic measurement networks - both for determining whether a proposed data set can constrain inversion and for optimizing monitoring networks comprised of multiple measurement types.

Applying Simple Machine Learning Tools to Infer Streambed Flux from Subsurface Pressure and Temperature Observations

Moghaddam, Mohammad[1], Ty P.A. Ferre[1], Xingyuan Chen[2], Kewei Chen[2], Xuehang Song[2], Glenn Hammond[3]

1 Hydrology and Atmospheric Sciences, University of Arizona, Tucson, AZ, USA

2 Pacific Northwest National Laboratory, Richland, WA, USA

3 Sandia National Laboratory, Albuquerque, NM, USA

Corresponding author: Ty Ferre (tyferre@gmail.com)

Key Points:

- Simple machine learning tools (regression tree and gradient boosting) can aid in the efficient design of hydrologic experiments
- Surface water exchange flux beneath a river can be determined with high temporal resolution (5 minute) using a single combined temperature and pressure probe
- High resolution surface water exchange flux beneath a river cannot be determined using temperature alone if temperature measurements include measurement error

Abstract

We demonstrate the application of two simple machine learning tools - regression tree and gradient boosting analyses - to a hydrologic inference problem to address two objectives. The first goal was to infer the flux between a river and the subsurface based on high temporal resolution (5-minute) observations of subsurface pressure and temperature. The second goal was to identify an optimal set of observations to support these inferences. Specifically, we examine how many and what type of observations (pressure and/or temperature) were necessary and at what depths. Using synthetic observations and surface fluxes provided by a fully resolved three-dimensional flow and heat transport model, we found that both machine learning tools could identify the flux well using pressure and temperature measurements collected at three depths, even when considerable noise was added to the synthetic observations. Neither method could provide reasonable flux estimates given only noisy temperature data. A shallow, collocated temperature and pressure observations performed as well as the complete data set. The results show the promise of using machine learning tools to design hydrologic measurement networks – both for determining whether a proposed data set can constrain inversion and for optimizing monitoring networks comprised of multiple measurement types.

Plain Language Summary

Machine learning has gained popularity for model-free interpretation of large data sets. We show that numerical models can be used to train machine learning tools, then those tools can be used to efficiently design optimal monitoring networks. This is applied to the task of inferring rapidly varying water exchange between the Columbia River and its underlying sediments.

1 Introduction

There is long-standing interest in developing methods to quantify surface water – ground water exchange flux to better understand water and solute exchange across the sediment-water interface. There has been particular interest in developing temperature-based methods, as presented in reviews by Anderson (2005), Constantz (2008), and Rau et al (2014, 2015). These methods are preferred because temperature sensors are generally less expensive and more robust than pressure sensors. In general, temperature-based methods have been based on inferring conductive-convective heat transport from time series of temperature at multiple depths to estimate water flux. Initial methods fit an analytical solution describing the subsurface response to a sinusoidally varying surface temperature forcing (Suzuki, 1960; Stallman, 1965). Later approaches have used numerical models to infer infiltration from temperature time series measured in the surface water and in the subsurface (e.g. Constantz et al, 2002). Several previous researchers have recognized that uncertainty in the sediment thermal parameters can translate to uncertainty in flux estimates (e.g. Constantz et al., 2003; Shanafield et al., 2011). But, to date, no published methods have considered estimating water flux under conditions of temporally varying, temperature-dependent hydraulic conductivity. Furthermore, the flux estimates have been limited to relatively low temporal resolution – hours to months – because the available data do not support unique inversion of the water flux boundary condition at a time resolution similar to the data collection frequency.

The exchange flux can be inferred from measured pressure and/or temperature through the development and use of a calibrated numerical model of flow and heat transport. This approach has the advantage that it can include known elements of the physical system, such as the temperature dependence of hydraulic conductivity, soil layering, and soil thermal properties. However, such an approach will be relatively computationally expensive to maintain over a long monitoring program, especially if multiple probes are installed. The goal of this study was to determine if a machine learning approach could leverage the effort spent on developing a numerical model to provide real time estimates of exchange flux with very low computational effort.

Our objective was to examine the potential uses of simple machine learning (ML) techniques to augment numerical model-based analyses of streambed infiltration/exfiltration. Specifically, we aimed to determine whether simple ML methods, trained on a numerical model, could provide near-real time flux estimations based on five-minute resolution subsurface pressure and temperature time series. Further, we examined whether the ML tools could be used to identify a reduced observation network, ideally comprised of only temperature sensors, that contained all information necessary to infer the surface/subsurface flux. If successful, ML tools could be paired with relatively few sensors to extend monitoring of water flux across the ground surface at low cost after an initial, more intensive calibration period.

2 Materials and Methods

This study represents an initial feasibility study of a novel use of ML in hydrology. Known time series of river stage and surface water temperature were used as inputs into a numerical flow and heat transport model. The numerical model produced exchange flux time series at the streambed as well as temperature and pressure time series at multiple depths. The numerical model output was used as input for the ML analyses. The exchange fluxes were the forecast targets and the temperature and water pressure time series at multiple depths were the features. That is, ML tools used temperature and pressure time series to infer the exchange flux time series at the streambed. In the following sections, we describe the numerical model, which was used to train the ML tools. Then we discuss how each of the ML tools used this common set of data and compare their performance for different observation sets. The ML methods are described in detail to provide an introduction to the use of these methods for hydrologists who may have less familiarity with their application. Readers with considerable familiarity with simple ML tools may choose to skip the ML methods sections.

2.1 Numerical Flow and Heat Transport Model

In this study, PFLOTRAN (Hammond et al., 2014) is used to simulate subsurface water flow and heat transport beneath a stream. PFLOTRAN is a parallel multiphase flow simulator implemented in object-oriented FORTRAN. For this study, PFLOTRAN was used to model fully-coupled nonisothermal flow and heat transport using an integral volume finite difference approach with the nonlinearities in the discretized equation resolved thorough Newton Raphson iteration.

A 1D model simulating flow and heat transport in vertical direction was built to generate synthetic temperature, pressure and flux data. The temperature and pressure were used as training data to infer flux and the inferred flux was validated against the simulated flux. The model is 2 m in length with a grid dimension of 0.01 m. High spatial resolution was necessary to increase the accuracy of the simulated results at depths selected for observations. The boundary conditions are Dirichlet type for both flow and heat transport. To reflect the possible complex conditions in the field, the time series of temperature and hydraulic heads that are assigned as boundary conditions were extracted from a 3D model. The 3D model was built to simulate flow and heat transport near the Columbia River at the Hanford 300A area in Washington. The model domain is 400 m × 400 m × 20 m, including three layers with different hydraulic and thermal properties (alluvium, Hanford and Ringold). The alluvium layer where hydrologic exchange flow occurs is the sole focus of this investigation. The grid dimension of the alluvium layer is refined to 0.5 m × 2 m × 0.1 m to capture the flow and temperature dynamics in the hyporheic zone. The model domain covers the location where a thermistor rod was installed in the field. The temperature and pressure from the surface and 2 m depth at the thermistor rod location in the model are used to drive the 1D model. The permeability of the alluvium layer is 3.86×10^{-11} m2. The simulation period was 1/1/2016-6/30/2017 (1.5 years) with temperature and pressure generated every 5 minutes.

The surface boundary conditions are the water pressure and temperature at the streambed through time. A larger scale, 3D model was run to generate boundary conditions for a high resolution, 1D model constructed for this study. The larger model included an uppermost layer with fine vertical resolution (0.1 m) that extended to 2 m depth. Beneath this surface alluvium layer, the grid cell size increased with depth to a maximum of 0.968 m. The bottom boundary for the larger model was 20 m below ground surface. A 1D vertical model was run for a 2 m vertical domain with 0.01 m cell resolution. The time series of values used to define the top and bottom Dirichlet boundary conditions for flow and transport were extracted from the results of the larger 3D model. Details for this larger scale model are available in (Bisht et al., 2017).

PFLOTRAN simulated water flow and heat exchange between the Columbia River and the underlying subsurface in three dimensions for 1.5 years with a time step of 5 minutes. The subsurface was assumed to be homogeneous within the top 2 m with a permeability of 3.86×10^{-10} m2. But, the hydraulic conductivity varied with the local water temperature due to the dependence of viscosity on temperature:

$$\mu_w = 241.4 * 10^{10247.8/(\mathrm{T}-140)}(1.0\ +\ 1.0467\ \times 10^{-6}\, 209\,(\mathrm{p}\ -\ \mathrm{p}_{\mathrm{sat}})(\mathrm{T}\ -\ 305)) \quad [1]$$

where μ_w is water viscosity in μP; T is temperature in K; p is pressure in bars and $\mathrm{p}_{\mathrm{sat}}$ is saturation the pressure in bars corresponding to temperature T (American Society of Mechanical Engineers, 1967).

2.2 Numerical Model Results Used for Machine Learning Analyses

The high-resolution 1D vertical flow and heat transport PFLOTRAN model generated pressure and temperature at 200 depths with a 0.01 m spacing between 0.005 and 1.995 m depth below the riverbed. We considered a subset of these measurement depths to represent a plausible monitoring network with sensors at 0, 0.015, 0.105 and 0.195 m depth: these depths represent a measurement atop the streambed, immediately below the surface of the bed, and two sensors placed at approximately 10 cm separation (Figure 1).

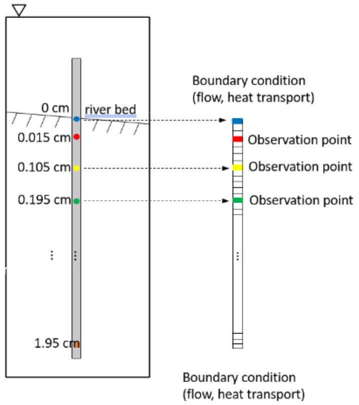

Figure 1) Location of observation points considered at the base of the water column and at three subsurface depths. Each location was considered to have a temperature sensor and/or a pressure sensor.

The objective of our application of ML was to infer the surface flux time series with 5-minute resolution (Figure 2A) from subsurface time series of pressure (Figure 2B) and/or temperature (Figure 2C). For the site conditions, flux is generally low; there are more instances of downward flux than upward and the maximum downward flux is greater than the maximum upward flux. (Note that downward flux is shown in Figure 2 as negative upward flux and that this convention is maintained throughout.)

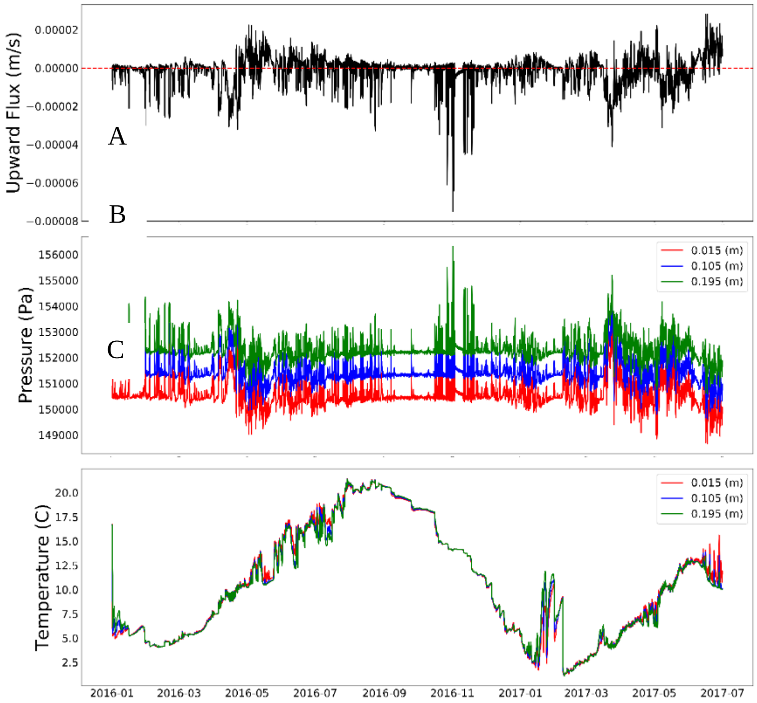

Figure 2) A: exchange flux calculated at the streambed, B: pressure at three depths, C: temperature at three depths

2.3 Training the Machine Learning Tools and Defining Observation Sets to Examine

In this study, we consider two machine learning methods. A key decision that is common to most ML analyses involves the definition of training and testing subsets of the data. Therefore, before discussing each ML method in detail, the following section explains how the data were divided between training and testing sets.

Because many ML tools allow for a very large number of degrees of freedom in fitting to data, it is critical to ensure that the ML fit reflects underlying relationships in the data rather than simply overfitting (memorizing) every variation in the training data. This is generally achieved by fitting the ML tool to some of the data and reserving some of the data for testing. For more challenging applications, it can be critical to form three data pools: training; validation; and testing. The additional validation pool is used for intermediate testing of results to avoid unintentional inclusion of information from the testing data during training. For this application, there is minimal tuning, so the definition of separate validation and testing sets was not deemed to be critical.

There is no hard rule for how to determine the amount of data to set aside for training. In general, simple ML tools such as those examined here see improved performance with more training data, but the performance reaches a maximum beyond which more training data do not improve performance (Zhu et al., 2015). In our case, we used 70% of the data for training to ensure that we included enough data to reach the maximum performance plateau. The training/testing sets could be formed by using the first 70% for training or by randomly assigning 70% of the data as training throughout the time series. The choice of the training/testing split should be objective, but there are some considerations to ensure reliable inference. Specifically, the trained ML tools used for this study can only interpolate among conditions on which they have been trained. So, for example, if the testing set includes temperatures that are higher or lower than the range of the training data, then these values will effectively be inferred to be equal to the highest/lowest value used in the training set. For our application, because the hydraulic conductivity depends on the temperature and the flux depends on the hydraulic conductivity and the pressure gradients, the training set had to be chosen to have a larger range of paired values of flux, temperature, and pressure than the testing set. This could not be achieved by defining the first 70% of the time series for training due to the timing of seasonal variations.

We faced another consideration in defining the training/testing sets. Namely, the exchange flux at time, t, will not necessarily be reflected in the pressure and temperature at depth at time, t. Rather, there may be delays in the propagation of pressure and temperature to depth. Therefore, we allowed the ML tools to consider temperature and pressure values at the time of surface flux inference and after some time delay. This consideration of time delayed observations was inconsistent with using random, nonconsecutive samples for training. That is, we had to ensure that time delayed observations were rarely drawn from the testing periods.

To sample the full range of temperature, pressure, and flux conditions while allowing for training on time-delayed observations, we divided the 110,000 observation times into six paired training/testing periods (Figure 3). Training was performed on observations: 500- 12500; 19000 - 25000; 33000 – 45000; 52000 – 70000; 75000- 90000; 97000- 110000 (shown in blue). Testing was performed on the remaining observations (shown in red). In a descriptive context, periods of maximum and minimum temperature, dictated by seasonal variations, were included in the training set. Meanwhile, periods of maximum upward and downward flow, dictated by both natural and management influences, were also included in the training period.

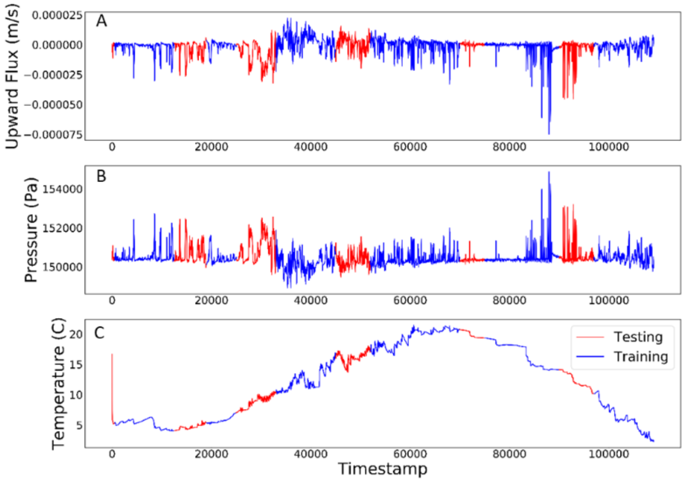

Figure 3) Time series of training (blue) and testing (red) sets illustrated using data from 0.005 m depth. A: upward flux, B: pressure, and C: temperature.

The numerical model produced error-free paired values of surface flux and subsurface temperature and pressure. Field observation will contain errors due to instrument limitations, uncertainties regarding subsurface placement, and many other unaccounted sources. It is difficult to know the true level of measurement uncertainty. Therefore, we applied a generic error model by adding zero mean Gaussian random errors with a standard deviation equal to a given percent of the variance of all measurements of that type collected at all depths and times. That is, the level of relative measurement error is assumed to be comparable for temperature and pressure sensors. But, while the errors are homoscedastic, they are calculated separately for each measurement type. To conform to general descriptions of measurement error, these errors are described as a signal to noise ratio (SNR). For example, if the variance of the error free observations is 100 times the applied variance of the added errors, then the SNR is reported as 100. A measurement set with an SNR of 10 would be considerably noisier than one with an SNR of 100.

In addition to considering temperature and pressure time series, we examined whether temporal and spatial gradients of temperature and pressure were more informative than direct measurements. For example, it would be expected that pressure gradients, which are directly related to vertical water flow, may be more informative than a pressure measurement at a single depth. To reflect practically achievable gradient observations, spatial gradients were calculated between the sensor depths already included in the observation set; that is, the addition of gradient measures did not increase the number of subsurface sensors needed. This restriction also applied when observation sets were downsampled in later analyses. We assumed that measurements were collected regularly, every 5 minutes, at any selected depth, so the temporal gradient at a given depth did not require additional observations.

In practice there is a strong preference for using only temperature measurements because temperature sensors are less expensive and more robust than pressure sensors. Therefore, one of the main objectives of our data-worth analyses was to determine whether pressure measurements were necessary for accurate flux assessment. To achieve this, we considered four measurement scenarios: including both pressure and temperature measurements at multiple depths; using only temperature measurements at multiple depths; using only a single pressure sensor; and using a single pair of collocated temperature and pressure observations, possibly provided by a single combined sensor.

2.4 Implementation of Regression Tree Analyses

Regression tree (RT) techniques consider paired values of targets (here, streambed exchange flux) and features (in this case, subsurface temperature and/or pressure observations and/or gradients). The analysis sequentially divides the fluxes (Figure 3A, training data) into subsets. At each point of division, the objective is to identify child subsets such that each has a lower variability of the flux values than the combined parent set. In our application, we use the mean squared error (MSE) between the mean value over the (sub)set and all members of the set as a measure of variability. Critically, the child sets must be divided based on defining a threshold value of one observation (e.g. pressure at a specific depth applied to all times). RT progresses by identifying the single observation and associated threshold value that results in the greatest reduction of variability of the child subsets at each successive node. In this way, RT is a ‘greedy’ algorithm: each identification of observation and threshold is made without regard to any future or past bases for segregation. For this reason, RT is not guaranteed to be optimally efficient. Rather, it is seen as a relatively simple, rapid, easily-interpreted ML approach.

RT produces a tree structure because each subdivided set is further separated. The number of subsets defined at each split point (node) and the number of levels of the RT are user-defined settings. Limits can also be placed on the minimum reduction in variability required and/or the minimum population of a subset needed to justify branching at a node. The number of levels, number of splits, and required variability reduction and/or subset size are hyperparameters that must be tuned for optimal RT performance.

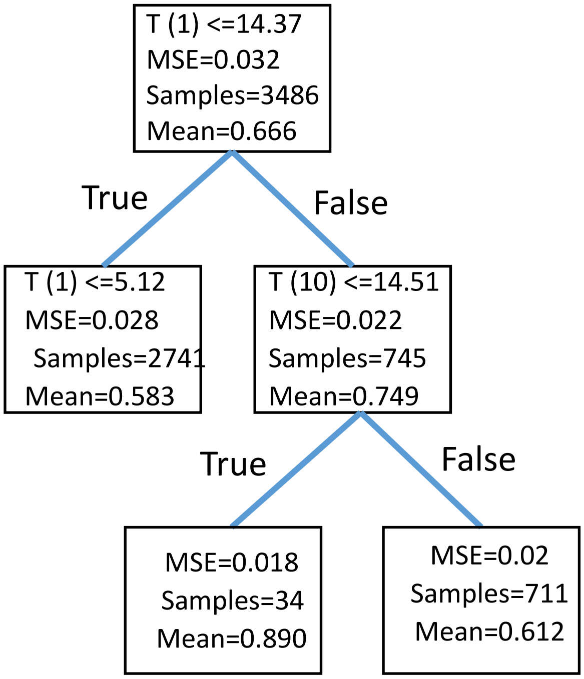

Figure4) Illustrative example of a two-level regression tree to segregate streambed exchange flux based on subsurface temperature observations at ten depths T(0), T(1) … T(10).

A strength of RT is its ease of interpretation. Consider the illustrative example shown in Figure 4. Temperatures measured at ten depths, T(1) through T(10), were considered to form a tree with only two children formed at each nodes and two levels of nodes. The initial set, including all of the training fluxes, was composed of 3486 samples with a mean value of 0.666 m/s; the MSE between all of the flux values and the mean is 0.032 m/s. The RT process identified the shallowest temperature observation, T(1), with a threshold of 14.37 ^{o}C as the first threshold for classification. This divides the fluxes into two groups with mean values of 0.749 and 0. 583 and sample sizes of 745 and 2741, respectively. The MSE values of the consequent groups are smaller than before splitting: 0.028 and 0.022. The sample-weighted MSE after splitting is 0.023 m/s. The left branch identifies T(1), again, as the best criterion with a threshold of 5.12 ^{o}C. But, this does not meet the minimum required improvement in MSE to continue, so the branching ends for this sequence. The right branch identifies T(10) with a threshold of 14.51 ^{o}C. This can be divided further into child subsets with sufficiently reduced MSE to justify a final branching. To apply the RT to a testing observation set, T(1) and T(10) would be measured. If, at time t, the measured values were T(1) = 15.0 and T(10) = 13.2, then the regression would follow the path to the central box with a mean value of 0.890: this mean value would be the inferred flux at the measurement time. That is, this simple RT would identify one of three values of flux based on two observations. Clearly, the resolution of the estimated flux depends on the number of divisions at each node and the number of layers. But, at some point further subdivision will not satisfy the minimum required improvement and further refinement will stop. As illustrated, the final resolution of flux depends on the data and the values assigned for the tuning parameters. As such, hyperparameter tuning also seeks to minimize overfitting by limiting the size of the tree to optimize performance on the testing set.

As described, RT would form the same tree every time it was applied to the same data set. While this repeatability or stability is useful for interpreting the results, it can also leave the method open to overfitting. That is, the tool can slavishly fit every detail of the data while missing some key underlying relationships. To avoid overfitting, a random element can be added such that the analysis only considers a subset of all available observations at each node. While this reduces overfitting, the analysis can give different results for repeat analysis of the same input data. We found only mild evidence of overfitting, so we did not employ this random selection in our analyses, resulting in stable RT results.

RT is very well suited to monitoring network design because it provides a clear map of the influence that different observations have on the interpretation of the target outcome. Considering the illustrative example, we could conclude that we are only using information from T(1) and T(10) to classify fluxes. Therefore, measurements T(2) through T(9) could be eliminated with no loss of accuracy. We can further assess the relative value of T(1) and T(10) by examining how much each reduces the variability of the resulting sets of estimated flux values. Specifically, at each node we can quantify the population-weighted reduction in MSE, thereby reflecting the number of samples affected by the observation and the degree of reduction of the variability at that node.

$$\gamma = \frac{1}{n_p} \sum n_i \left(MSE_p - MSE_i \right) \qquad [2]$$

where n_i is the number of observations in observations in the subset, MSE_i is the mean squared error of that subset, np is the number of observations in the parent set, and MSEp is the mean squared error of the parent set. These nodal importance values can be summed for each observation (e.g. for all instances of T(1)) and then normalized by the sum over all observations to define the relative contribution of each observation:

$$i_J = \frac{\sum_{j\epsilon J} \gamma_j}{\sum_{k\in K} \gamma_k} \qquad [3]$$

in which J is the set of all nodes considering observation j, and K is the set of all nodes.

2.5 Implementation of Gradient Boosting Analyses

RT is a conceptually accessible ML method. But, it has several well-recognized limitations (Hastie, 2005; Prasad et al., 2006). As described above, RT can be susceptible to overfitting, especially if the input data are noisy. The 'greedy' nature of RT can miss combinations of splitting rules that may lead to better segregation, but cannot be identified sequentially. Additionally, because each leaf is represented by the mean value of its samples, the model has a stepped output, especially for trees with relatively few levels. Finally, RT is a relatively inefficient algorithm. Specifically, the algorithm expends computational effort to attempt to subdivide every node on every level whereas it would be much more efficient to expend more effort on areas where the RT is underperforming based on the previously defined levels.

One conceptually simple, but powerful, approach to address the limitations of RT is to use an ensemble method. To understand ensemble methods, one can imagine conducting multiple RTs and averaging their responses. Each contributing RT would be relatively shallow (a few layers); as such, it is referred to as a 'weak learner'. Clearly, something has to ensure that the contributing trees are different; this can be achieved by only showing each tree (or each node within each tree) some fraction of the training set, as described above. By shuffling the available paired observations of target and data, an ensemble method allows for observation combinations to be included that would not be found by requiring sequential selection of the next single best observation. The inferences (and importance values) of the independent tress can be averaged, giving a more robust inference that is less susceptible to overfitting. The potential down-side of ensemble methods is that they can be relatively inefficient, requiring the training of many trees that may not contribute much to the final inference.

The preceding approach to forming an ensemble is referred to as bagging (bootstrap aggregating); a more efficient approach is known as boosting. Boosting begins by constructing a weak tree (Schapire et al. 1990). Then, the next tree is constructed to address the uncertainties that remain (the residuals) after applying the first tree. Gradient boosting (GB) is an example of a boosting ensemble tree-based ML approach that has been used widely for both regression and classification problems (Touzani et al., 2018; Wei et al., 2019; Huang et al., 2019; and Ransom et al., 2017).

To understand how GB would be applied to the illustrative example shown above for RT, we can consider defining a second two level tree. The first tree, as shown above, was developed to reduce the variance of the flux values by subdividing them into bins. The second tree would consider the difference between each known flux value and the value assigned to it by the first tree. Each successive tree is developed to address the difference between the known target fluxes and the estimated fluxes based on all previous trees. To use the trained GB, the path through the first tree would be followed based on the identified observations and thresholds. The mean value in the terminal bin would be the first estimate of the flux at that time. Then, the path through the second tree would be followed. The mean of the resulting bin would be added to the first estimate. Each subsequent tree contributes further additive corrections to the flux estimates. The degree of improvement of fit provided at each node in each tree defines the importance of each observation.

In this way, the importance of each observation is calculated over all trees in the GB in a manner conceptually similar to that for RT. GB is a much more efficient use of computational resources than RT because the algorithm is continually focused on addressing the remaining mismatch between the training value and the inferred value based on the analysis to that point. But, GB is also susceptible to overfitting: given enough sequential trees, the algorithm will fit the noise in the training set. To reduce overfitting, each successive tree is weighted less than the previous tree in the sum using a variable known as the learning rate; that is, only a fraction of the proposed correction is applied for each tree.

2.6 Hyperparameter Tuning

The purpose of this investigation is to illustrate the application of these two relatively simple ML tools, RT and GB, to a hydrologic inference challenge. Specifically, we examine the performance of the two approaches for inferring infiltration/exfiltration as well as their ability to identify informative reduced observation sets. One subobjective is to compare the performance of the methods; this requires that each is tuned to get the best performance. Specifically, both RT and GB require a user to define the hyperparameter values to optimize their performance. RT requires definition of the number of splits per node, the number of levels, and the minimum threshold for performing a split. In addition, RT can allow for a user-defined fraction of the observations to be considered at each node to protect against overfitting. GB requires the user to define the structure of each tree, as for RT, the number of trees and the learning rate. The process of optimizing an ML tool to produce the best possible inferences while avoiding overfitting is known as hyperparameter tuning. There are few definitive guides for choosing hyperparameter values. Rather, it is common to use approaches like cross validation, wherein the training data are partitioned into several training/testing subsets (Arlot and Celisse, 2010). Each of these subsets is assessed and both the average performance and the variation of the performance across the different splits is examined; the tuning parameters are adjusted to improve one or both of these measures. We used five-fold cross validation and a standard grid search to tune the hyperparameters for RT and GB. The resulting optimized parameter values are reported for each analysis in Table 1.

Analysis	n_estimators	max_depth	learning_rate	min_samples to split	min_var reduction to split	Dataset	Noisy
RT		7			0.001	P and T	TRUE
RT		7			0.001	P and T	FALSE
RT		20			0.001	only T	FALSE
RT		12			0.001	only T	TRUE
RT		7			0.001	one P	TRUE
RT		7			0.001	one P	FALSE
RT		12			0.001	one P one T	FALSE
RT		7			0.001	one P one T	TRUE
GB	1000	5	0.05	40		P and T	FALSE
GB	1000	5	0.1	20		P and T	TRUE
GB	1000	5	0.05	40		one P one T	FALSE
GB	1000	5	0.05	100		one P one T	TRUE
GB	1000	10	0.1	20		only T	FALSE
GB	200	10	0.008	40		only T	TRUE
GB	1000	3	0.008	500		one P	FALSE
GB	1000	3	0.008	500		one P	FALSE

Table 1) Tuned hyperparameter values for each ML application

3 Results and Discussion

To illustrate the application of RT and GB to flux inference based on model-derived subsurface temperature and pressure data, we applied each method to common pressure and/or temperature time series and assessed their performance based on the accuracy of the exchange flux estimations for the test set. To examine the optimal composition of a sensor network to quantify flux with five-minute resolution, we repeated these analyses with temperature and/or pressure time series measured at different depths as well as considering time-delayed observations and spatial and temporal gradients. For practical considerations, we specifically considered temperature-only data sets. Finally, to examine the impact of measurement error on the ML performance, we repeated these analyses with and without added measurement error. The results for each method are presented in the following order to meet these objectives. First pressure and temperature data are considered at all observation depths, with and without error; the performance is assessed, and optimal observation sets are identified. Then, the same analyses are presented using only temperature observations at all depths. Finally, we considered pressure and temperature with the restriction that measurements could only be made at a single, common depth to represent the use of a combined sensor.

3.1 Regression Tree – Temperature and Pressure

Initially, RT was applied with a robust data set, including both temperature and pressure measurements at 0.005 (representing water in the stream), 0.15, 0.105, and 0.195 m depth. Pressure and temperature measurements were considered at the time of flux inference and 20 minutes later. (Other time delays, between 5 and 30 minutes showed similar results. For clarity, only results for the 20-minute delay are shown.) In addition, the temperature difference between these times was considered; these are referred to as temporal gradients (dt). Depth differences were considered for both observation types. These depth differences were calculated at the same times as the pressure/temperature measurements and with a 10-minute delay. These are referred to as spatial gradients (dz).

For both the training and testing sets, the RT was able to infer upward fluxes very accurately (Figure 5A and 5C) with no added noise. Downward flux was less well resolved. More specifically, both upward and downward flux were estimated accurately for fluxes less than approximately 0.00002 m/s; higher values of downward fluxes were less well resolved. This is somewhat surprising, given temperature-based estimation is based on inferring water flux from advective heat transport and that advection should be more pronounced compared to diffusion for higher water flux conditions. One possible explanation is that the higher fluxes are less common, leading to less refined inference using the RT method. This is supported by the relatively wide ranges of fluxes that are interpreted as a single value: horizontal series of points on Figure 5. The similarity of the fit for training and testing indicate very little overfitting. For lower fluxes, there was only minor reduction of the quality of the estimated flux due to added noise with an SNR of 100(Figures 5 B and 5D). However, adding noise further degraded the quality of the estimates of high downward fluxes.

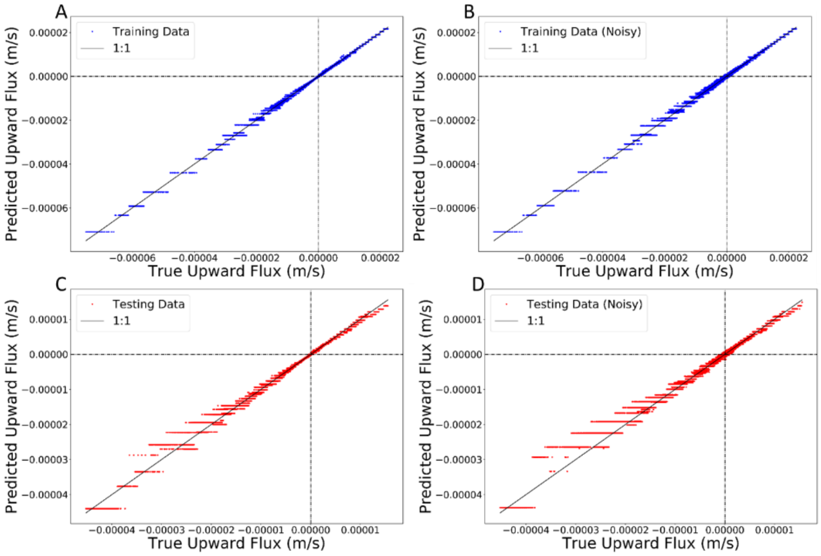

Figure 5) Training and testing results using temperature and pressure sensors, which are located at, 0.015, 0.105, 0.195 m. A: training set fit for noise free data. B: training set fit for SNR=100 noisy data. C: testing set fit for noise free data. D: testing set fit for SNR=100 noisy data.

As described, RT allows for relatively simple interpretation of the observations (types, depths, delays, or gradients) that contribute to the regression. In reality, the specific choice of optimal observations may depend on the specific error realization, which cannot be known at the time of network design. Therefore, we examined the general impact of noise on the design of a monitoring network by running 100 error realizations and averaging the feature importance values. Each realization had the same SNR, but different specific error time series added to the observations.

Considering all of the pressure and temperature observations, we find some commonalities and differences in the observations identified based on whether noise has been added to the data (Figure 6). With no noise added (blue bars), the most informative observation was a deep pressure gradient With noise added (orange bars), the most informative observation was a shallow pressure measurement. This is likely due to the homoscedastic nature of the errors (favoring larger contrasts in observations). Similarly, differencing of noisy data amplifies measurement errors because the error of a sum or difference of observations is the square root of the sum of the squares of each observation error. Despite these qualifiers, it is somewhat unexpected that deep gradients were preferred so strongly for the error-free observations.

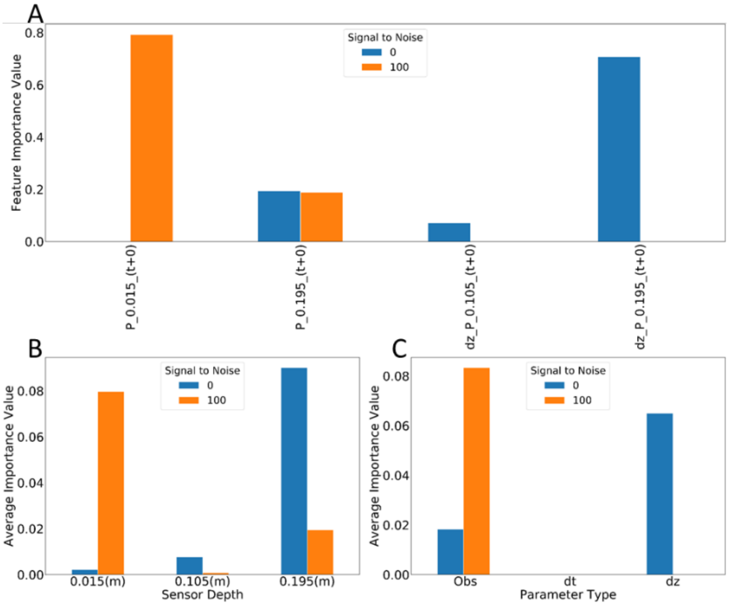

Figure 6) A: feature importance for pressure and temperature observations (P and T) and spatial (dz) and temporal (dt) gradients for sensors at, 0.015, 0.105, 0.195 m depth with (orange) and without (blue) measurement error. The time delay after the flux estimation is shown in parentheses. B: summary of feature importance by depth. C: summary of feature importance by type – observed value, temporal gradient (dt), and spatial gradient (dz).

3.2 Regression Tree – Temperature and Pressure

There is a strong practical advantage of using temperature sensors rather than relying on pressure sensors (Anderson, 2005; Constantz, 2008; Rau et al., 2014, 2015). But, because pressure observations were preferred with both pressure and temperature were considered and RT is a greedy algorithm, the only way to assess the information content of temperature observations was to remove the pressure observations from consideration. The results show that an RT can be trained reasonably well on temperature data with no error (Figure 7A), although high downward fluxes are still poorly resolved. For temperature only, with sensors placed at all three depths, an error with an SNR of 100 resulted in very poor inference of flux based on a trained RT (Figure 7B). This does not explicitly indicate that temperature observations do not have enough information to constrain flux estimation. Rather, it shows that an RT cannot be trained successfully on the information available in temperature observations alone.

The clear difference in performance with and without noise (Figures 7A and 7B) highlights the well-known sensitivity of RT methods to noisy data (Kerdprasop et al., 2011). We applied two approaches to mitigate the effects of noise; both approaches aimed to infer flux with lower temporal resolution (30 minute averages). First, we estimated the flux with 5-minute resolution and then averaged the results as non-overlapping 30-minute windows. Second, we averaged the true flux into non-overlapping 30-minute values and trained the RT using corresponding 30-minute average temperature values. Neither approach was successful.

Based on our inability to infer flux using temperature only, we considered a single pressure sensor, placed at 0.105 m depth, to be the next most practically-preferred design. Given that we only considered a single sensor, we could not consider spatial gradients. The results (Figure 7C and 7D) show that an RT can be trained well on a single pressure observation. Furthermore, the results are relatively insensitive to measurement error. In all cases, the absolute pressure was favored over temporal gradients. This is a practically promising result, as it may be more manageable to install and maintain a single pressure sensor in the field. The results also demonstrate the ability of RT to use time-varying information to make inferences; without the use of RT, it would not be possible to infer flux directly from a single pressure measurement even under conditions of temporally constant hydraulic conductivity.

Our final analysis considered the use of a combined pressure/temperature probe placed at 0.105 m depth, represented by time series of pressure and temperature at the same depth. It was not possible to use depth gradients using a single probe, but a 20-minute time difference was considered for temperature and pressure. An RT shows that the addition of temperature (Figure 7E and 7F) offers some improvement over using pressure alone (Figure 7C and 7D), especially for the relatively poorly resolved high downward fluxes. In fact, considering noisy conditions, the performance of a single combined sensor (Figure 7F) is as good as allowing the RT to consider all temperature and pressure data (Figure 5D). No time differences were identified as important. For the single combined sensor case, absolute pressure contributes 98% of the information; but, the 2% contributed by temperature improves the flux inference. We suspect that this improvement is due to the temperature observation helping to resolve the temperature dependence of hydraulic conductivity.

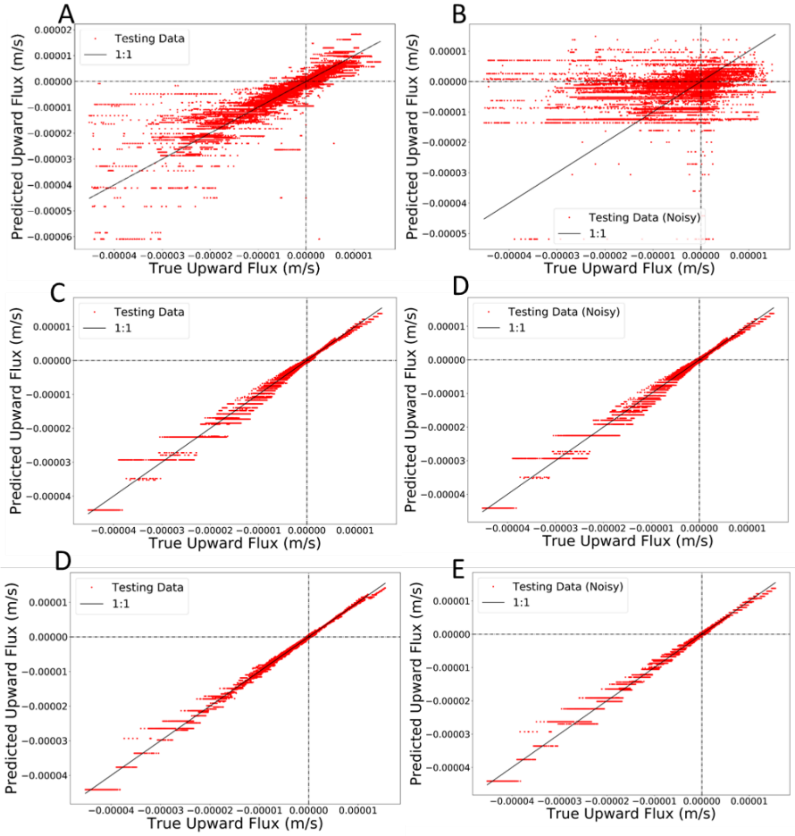

Figure 7 Predicted vs observed fluxes for the training set with noise free data on the left and noisy data (SNR=100) on the right. A,B: temperature sensors at all three depths. C,D: one pressure sensor at 0.015m depth. E,F: collocated pressure and temperature sensors at 0.015m.

3.3 Gradient Boosting Analysis

Gradient boosting was applied to the same data sets described above for RT. The GB results (Figure 8), are qualitatively similar to those for RT (Figures 5 and 7). Namely, using pressure and temperature at multiple depths, GB was able to infer the flux with high accuracy (Figure 8a and b). Using only temperature data appears to provide reasonable estimates of flux with no noise (Figure 8c), but the performance is highly degraded when measurement noise is added (Figure 8d). As was found when using RT, performance of GB was not improved significantly when estimating the 30-minute average flux (not shown). A single pressure sensor provides high quality flux estimates (Figures 8e and 8f), which are improved noticeably by the addition of a collocated temperature sensor (Figures 8g and 8h). This single combined sensor performs as well as an array of multiple sensors (Figures 8a and 8b). It is worth noting that GB seems to mitigate the poor estimation of high downward flows by RT (Figures 7 and 8). This provides an example for which the use of a more sophisticated ML algorithm is warranted because it can extract more information from the available data. In contrast, it would appear that RT is likely sufficient to interpret a single collocated pair of temperature and pressure sensors. It is possible that an even more sophisticated ML would be able to extract fluxes from temperature-only data; but, this may come at the expense of the interpretability of the simple ML tools presented here.

As for the RT analysis, when allowed to consider all available observations, GB selected deep pressure gradients as most informative for error free conditions (Figure 9). Of course, designs for a real field network must consider measurement error. When errors are added, GB selects all of the three direct pressure observations, with decreasing weight with depth. It also selects (with low weight), pressure measurements collected 20 minutes after the flux estimation time. In general, this indicates that GB is able to look more broadly at sources of potential information than RT. But, it is also worth noting that flux can be interpreted almost as accurately using only a single collocated pair of pressure and temperature sensors (Figure 8b versus 8h). This highlights the need to interpret ML-based observation selections critically, forcing the consideration of specific designs such as a single collocated pressure and temperature observation.

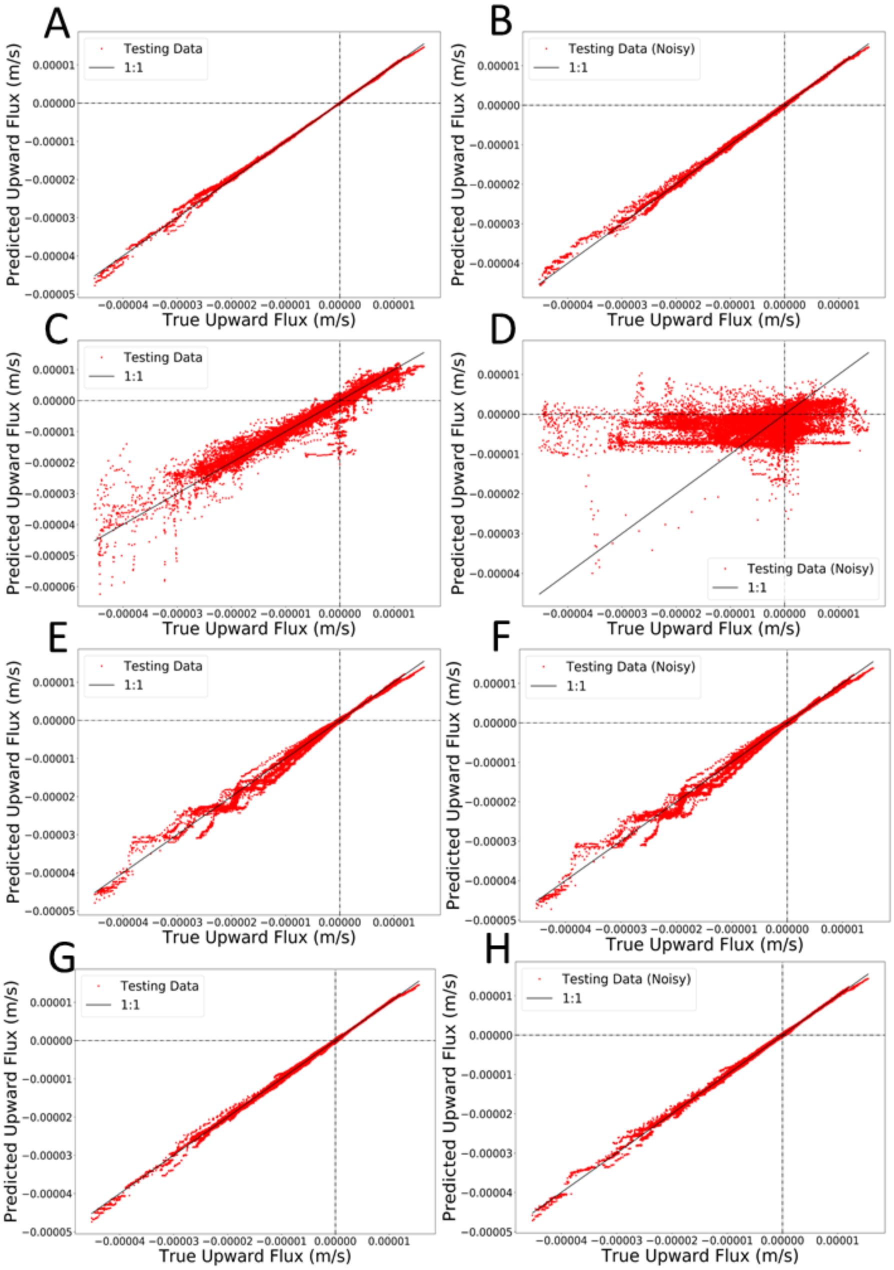

Figure 8) Testing set predicted vs observed fluxes. For multi sensor scenarios, sensors planted at 0.015, 0.105, 0.195 m. Sensors in the single case scenarios were located at 0.015m depth. The left panels show results for noise free data while the right panels represent noisy data. A,B: All-data; C,D: only temperature; E.F: single pressure; G,H: single collocated pressure and temperature.

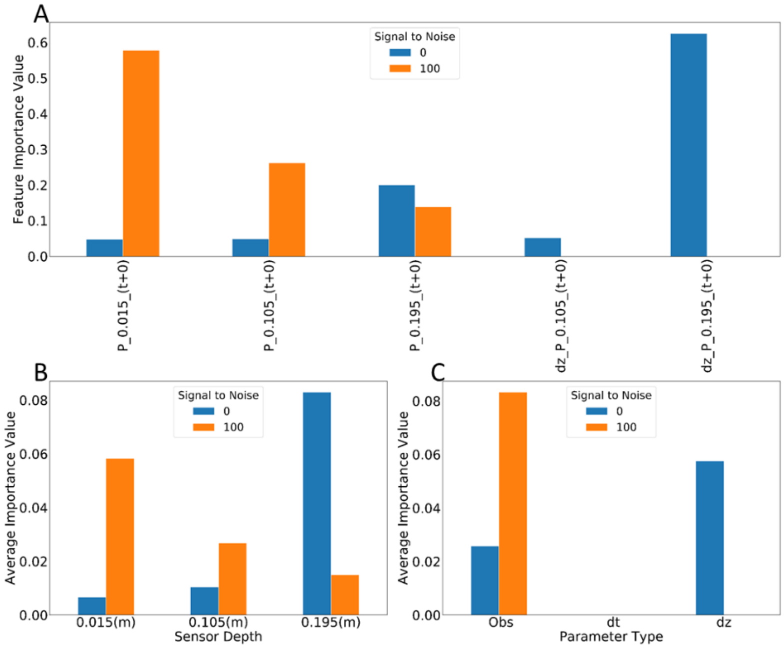

Figure 9).Gradient boosting feature importance for pressure and temperature sensors planted at, 0.015, 0.105, 0.195 m.

3.4 Optimizing Single-Sensor Arrays

Our ultimate aim was to investigate the potential use of two simple ML tools to infer high temporal resolution streambed infiltration from subsurface temperature and pressure measurements. Generally, we found that a single pressure sensor or a collocated single pressure and temperature set performed as well as a multi-sensor network. That leaves the choice of the depth at which to place these sensors. Further investigation (Figure 10) showed that the performance – based here on the R^2 between the actual and inferred fluxes for the testing set – was uniformly improved by choosing a GB analysis over RT, but the improvement was relatively small. In fact, the use of GB rather than RT offered approximately the same improvement as adding a temperature sensor to a single-pressure-sensor observation set. Of practical importance, any sensor depth above approximately 1 m was acceptable, with performance degrading considerably for deeper sensors. This insensitivity to the specific depth of deployment could be particularly important for highly dynamic riverbed conditions.

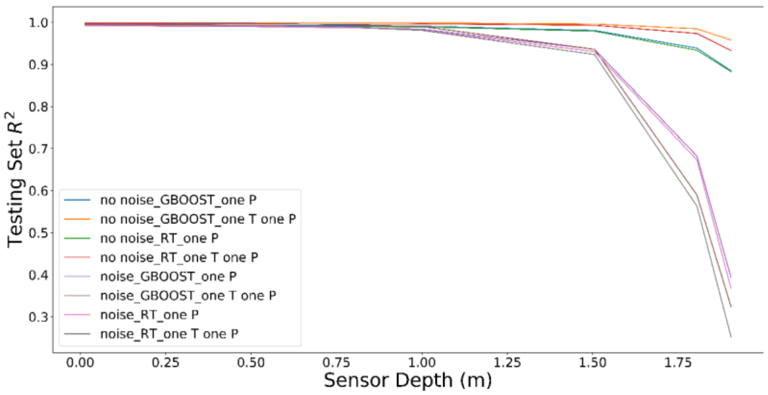

Figure 10) Performance of one pressure sensor and a combined pressure and temperature observation set with respect to the depth of installation and the ML tool used for analysis.

4 Conclusions

We demonstrate the application of two simple machine learning (ML) tools to a hydrologic inference problem. The primary goal was to provide high resolution (5-minute) estimations of flux between a river and the subsurface based on correspondingly high-resolution subsurface pressure and temperature observations. The secondary goal was to identify an optimal set of observations to support these inferences. We found that both regression tree (RT) and gradient boosting (GB) analyses could support these inferences given pressure and temperature measurements collected at three depths. The results were robust when noise was added to the observations. Using temperature data only, neither method could provide reasonable flux estimates when subject to measurement noise. Finally, a single collocated temperature and pressure observation, perhaps provided by a single sensor, performed as well as the complete data set. The depth of this combined sensor was not critical, as long as it was relatively shallow (< 1m).

The results show the promise of using machine learning tools to support hydrologic investigations. The advantage of using ML-based analyses is clear: a single combined pressure and temperature probe could be interpreted using a simple algorithm to provide real time estimates of exchange flux. But, several cautionary notes are raised as well. Namely, the ML tools investigated are not designed to consider monitoring network cost or complexity; therefore, these considerations have to be introduced intentionally in the analyses. Second, these methods can be sensitive to measurement error; this was especially evident when attempting to use only temperature observations to infer streambed flux. All efforts to reduce the impact of noise on this inference task were unsuccessful. In contrast, the results were largely insensitive to noise added to pressure observations. This difference deserves further consideration to draw general conclusions regarding the suitability of ML tools for specific hydrologic applications.

5 References

American Society of Mechanical Engineers (1967), Thermodynamic and transport properties of steam, p.75, New York.Anderson, M. P. (2005). Heat as a ground water tracer. Groundwater,43,951–968. https://doi.org/10.1111/j.1745-6584.2005.00052.x

Arlot, Sylvain, and Alain Celisse. (2010) A survey of cross-validation procedures for model selection. Statistics surveys 4: 40-79.

Chen, K., X. Chen, X. Song, M.A. Briggs, P. Jiang, P. Shuai, G. Hammond, H. Zhan and J.M. Zachara. (in preparation) Using ensemble data assimilation to estimate transient hydrologic exchange fluxes under highly dynamic flow conditions.

Constantz, J. (2008). Heat as a tracer to determine streambed water exchanges. Water Resources Research,44. https://doi.org/10.1029/2008wr006996

Constantz, J., Cox, M.H., Su, G.W. (2003) Comparison of heat and bromide as groundwater tracers near streams. Ground Water 41, 647–656.

Constanz, J., AE Stewart, R. Niswonger, and L Sarma. (2002) Analysis of temperature profiles for investigating stream losses beneath ephemeral channels. Water Resources Research, 38, DOI: 10.1029/2001WR001221.

Hammond, Glenn E., P.C. Lichtner, and R.T Mills. (2014) Evaluating the performance of parallel subsurface simulators: an illustrative example with PFLOTRAN. Water Resources Research 50.1.

Hastie, Trevor, Robert Tibshirani, and Jerome Friedman. (2005) The elements of statistical learning: data mining, inference and prediction. The Mathematical Intelligencer 27.2: 83-85.

Huang, Guomin, Lifeng Wu, Xin Ma, Weiqiang Zhang, Junliang Fan, Xiang Yu, Wenzhi Zeng, and Hanmi Zhou. (2019) Evaluation of CatBoost method for prediction of reference evapotranspiration in humid regions. Journal of Hydrology 574: 1029-1041.

Kerdprasop, Nittaya, and Kittisak. (2011) A Heuristic-Based Decision Tree Induction Method for Noisy Data. Database Theory and Application, Bio-Science and Bio-Technology. Springer, Berlin, Heidelberg: 1-10.

Prasad, Anantha M., Louis R. Iverson, and Andy Liaw. (2006) Newer classification and regression tree techniques: bagging and random forests for ecological prediction. Ecosystems 9.2: 181-199.

Ransom, Katherine M., Bernard T. Nolan, Jonathan A. Traum, Claudia C. Faunt, Andrew M. Bell, Jo Ann M. Gronberg, David C. Wheeler, Celia Z. Rosecrans, Bryant Jurgens, Gregory E.

Schwarz, Kenneth Belitz, Sandra M. Eberts, George Kourakos, and Thomas Harter. (2017) A hybrid machine learning model to predict and visualize nitrate concentration throughout the Central Valley aquifer, California, USA. Science of the Total Environment 601: 1160-1172.

Rau, G. C., Andersen, M. S., McCallum, A. M., Roshan, H., and Acworth, R. I. (2014). Heat as a tracer to quantify water flow in near-surface sediments. Earth-Science Reviews,129,40–58. https://doi.org/10.1016/j.earscirev.2013.10.015

Rau, G. C., Cuthbert, M. O., McCallum, A. M., Halloran, L. J. S., and Andersen, M. S. (2015). Assessing the accuracy of 1-D analytical heat tracing for estimating near-surface sediment thermal diffusivity and water flux under transient conditions. Journal of Geophysical Research - Earth Surface.120, 1551–1573. https://doi.org/10.1002/2015JF003466

Schapire, Robert E. (1990) The strength of weak learnability. Machine learning 5.2: 197-227.

Shanafield, M., Hatch, C., Pohll, G. (2011) Uncertainty in thermal time series analysis esti-mates of streambed water flux. Water Resour. Res. 47, W03504.
Stallman, R.W. (1965) Steady one-dimensional fluid flow in a semi-infinite porous medium with sinusoidal surface temperature. J. Geophys. Res. 70, 2821–2827.

Suzuki, S. (1960) Percolation measurements based on heatflow through soil with special reference to paddy fields. J. Geophys. Res. 65, 2883.

Touzani, Samir, Jessica Granderson, and Samuel Fernandes (2018) Gradient boosting machine for modeling the energy consumption of commercial buildings. Energy and Buildings 158: 1533-1543.

Wei, Zushuai, Yizhou Meng, Wen Zhang, Jian Peng, and Lingkui Meng (2019) Downscaling SMAP soil moisture estimation with gradient boosting decision tree regression over the Tibetan Plateau. Remote Sensing of Environment 225 (2019): 30-44.

Zhang, Yanru, and Ali Haghani (2015) A gradient boosting method to improve travel time prediction. Transportation Research Part C: Emerging Technologies 58: 308-324.

Zhu, X. et al. (2015) Do we Need More Training Data? https://arxiv.org/abs/1503.01508

6 Acknowledgments

This research was supported by the U.S. Department of Energy (DOE), Office of Biological and Environmental Research (BER), as part of BER's Subsurface Biogeochemical Research Program (SBR). This contribution originates from the SBR Scientific Focus Area (SFA) at the Pacific Northwest National Laboratory (PNNL). PNNL is operated for the DOE by Battelle Memorial Institute under contract DE-AC05-76RL01830. Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of

Energy’s National Nuclear Security Administration under contract DE-NA0003525. This research used resources of the National Energy Research Scientific Computing Center, a DOE Office of Science User Facility supported by the Office of Science of the U.S. Department of Energy under contract DE-AC02-05CH11231. This paper describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the paper do not necessarily represent the views of the U.S. Department of Energy or the United States Government.

7 Data Availability:

The data that supports the findings of this study are openly available in Center For Research Data at https://doi.org/10.4121/uuid:c874327b-9d70-4fd4-b96c-478eebd8ba21. This material was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor the United States Department of Energy, nor Battelle, nor any of their employees, nor any jurisdiction or organization that has cooperated in the development of these materials, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness or any information, apparatus, product, software, or process disclosed, or represents that its use would not infringe privately owned rights.

Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof, or Battelle Memorial Institute. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.