TECHNICAL DOCUMENTATION

UNIVERSITY OF CALIFORNIA
Santa Barbara

REPRODUCIBLE MACHINE LEARNING APPROACH FOR INTERPRETING ECOHYDROLOGIC MODEL OUTPUTS

A Capstone Project submitted in partial satisfaction of the requirements for the degree of
Master of Environmental Data Science
for the
Bren School of Environmental Science & Management

by

ALEX CLIPPINGER
WYLIE HAMPSON
SHALE HUNTER
PETER MENZIES

Committee in charge:
DR. ALLISON HORST
DR. NAOMI TAGUE

JUNE 2022

TABLE OF CONTENTS

SIGNATURE PAGE

REPRODUCIBLE MACHINE LEARNING APPROACH FOR INTERPRETING ECOHYDROLOGIC MODEL OUTPUTS

As developers of this Capstone Project documentation, we archive this documentation on the Bren School's website such that the results of our research are available for all to read. Our signatures on the document signify our joint responsibility to fulfill the archiving standards set by the Bren School of Environmental Science & Management.

_____

ALEX CLIPPINGER

_____

WYLIE HAMPSON

_____

SHALE HUNTER

_____

PETER MENZIES

The Bren School of Environmental Science & Management produces professionals with unrivaled training in environmental science and management who will devote their unique skills to the diagnosis, assessment, mitigation, prevention, and remedy of the environmental problems of today and the future. A guiding principle of the School is that the analysis of environmental problems requires quantitative training in more than one discipline and an awareness of the physical, biological, social, political, and economic consequences that arise from scientific or technological decisions.

The Capstone Project is required of all students in the Master of Environmental Data Science (MEDS) Program. The project is a six-month-long activity in which small groups of students contribute to data science practices, products or analyses that address a challenge or need related to a specific environmental issue. This MEDS Capstone Project Technical Documentation is authored by MEDS students and has been reviewed and approved by:

_____

DR. ALLISON HORST

_____

DR. NAOMI TAGUE

_____

DATE

ABSTRACT

The use of machine learning algorithms for predictive modeling is a growing area of study in the fields of ecology and hydrology. However, these methods have not been fully utilized to investigate process-based ecohydrologic model output. The purpose of this Capstone Project is to develop a framework that models, summarizes, and visualizes important variable relationships within data produced by the Regional Hydro-Ecologic Simulation System (RHESSys)—an ecohydrologic model designed by the Tague Team Lab at Bren. One of two machine learning techniques, random forest and gradient boosting, is used to rank variables by their importance in predicting a chosen response variable—this offers users a better understanding of where to focus their research. The primary deliverables of this project include a reproducible workflow with extensive documentation on necessary data preparation and machine learning concepts—as well as an interactive application to view workflow results and explore relationships within data. Using these tools, researchers can more efficiently analyze, explore, and identify important variable relationships in RHESSys datasets.

EXECUTIVE SUMMARY

Ecohydrologic models are core tools when investigating how climate can influence water, carbon, and energy cycles in natural and urbanizing landscapes. The Regional Hydro-Ecologic Simulation System (RHESSys) is a GIS-based ecohydrologic model that assesses nutrient and water cycling at varying spatial and temporal scales. RHESSys and other similar models are becoming increasingly sophisticated due to advances in software engineering, increased data availability, and a growing body of knowledge within the field of ecohydrology. As a result of this increased complexity, interpreting the large volume of data they output is a key challenge for both scientists and the public. Machine learning algorithms have the capability of quantifying patterns in data that are not revealed by traditional statistical analysis, and have been successfully implemented in ecology (Thessen 2016). A 2021 study conducted by Burke et al. demonstrated the viability of using random forests to analyze RHESSys output. Despite this, there remains limited guidance on employing machine learning methods in ecohydrology, requiring collaboration between computer scientists and earth scientists (Guswa et al. 2020). As a result, the vast majority of studies using RHESSys output to analyze watershed processes are only assessed using statistical summaries and time-series data visualizations.

This project provides a reproducible workflow that applies machine learning techniques to RHESSys output in order to reveal important variable relationships—it is intended to expedite data exploration and guide further analysis . This will help the RHESSys model become more accessible to researchers, which can lead to more robust data for watersheds globally. In addition, detailed documentation can help bridge the gap between scientific research and machine learning by providing a means by which salient effects can be efficiently extracted from complex model output, which would otherwise be uninterpretable for anyone without a high level of domain expertise. Tague & Frew (2021) have also identified a need for better visualization of hydrologic model output, which the Capstone Team has addressed through the creation of an interactive web application. This supports the broader goals of spurring involvement in local natural resource protection and creating a deeper connection between society and the natural systems that we depend on.

PROBLEM STATEMENT

Climate change presents a significant challenge to forest ecosystems around the world. Understanding the relative impacts of climate change on ecohydrologic variables is vital for management and conservation practices moving forward. RHESSys, a process-based ecohydrologic model capable of simulating water and nutrient cycling at varying spatial and temporal scales, is a powerful tool for researchers to predict these impacts (Tague and Band 2004). Recent RHESSys applications include investigating fuel treatment effects on forest health and fire vulnerability (Burke et al. 2021), the effects of timing of precipitation and snowmelt recharge on streamflow, and more (Tague and Peng 2013). However, extracting meaningful patterns from the large volume of highly dimensional output is challenging and stands as a barrier to accessing the model's full potential. For example, RHESSys output may consist of categorical variables such as climate scenario, drought condition, topography, basin, or stratum, as well as dozens of ecohydrologic variables such as leaf area index, net primary productivity, and root zone soil storage—resulting in a single dataset that can include daily observations of these variables spanning decades. Machine learning techniques can help researchers make sense of such data by quickly identifying important variables and interactions to expedite the data exploration process. There is a need for easily reproducible methods for applying machine learning to RHESSys output, as no such framework currently exists. For this Capstone Project, the team has developed a series of reproducible workflows that apply machine learning techniques to identify trends and patterns in RHESSys output data, as well as a Shiny application for interactive visualization of results.

OBJECTIVES

The overall success of the Capstone Project is measured through the following objectives.

1. Create reproducible workflows to help new users apply machine learning techniques to extract meaningful ecohydrological relationships from RHESSys output data.
2. Test the workflow implementation for RHESSys output data from the Sagehen Creek watershed.
3. Construct an interactive web application that decision-makers and the public can use to visualize and communicate relationships between climate change and watershed health derived from model output.

SUMMARY OF SOLUTION DESIGN

Approach and Methods

The Capstone Project utilizes RHESSys simulation output for the Sagehen Creek Experimental Forest. This site is on the Eastern slope of the Northern Sierra Nevada and encompasses varying elevation and climate. This initial case study is supplied as guidance for researchers to apply this workflow to their own research projects on other RHESSys watersheds.

The Capstone Team used the Sagehen Creek data to incrementally test and develop the workflow. This process began by developing a means of identifying the relative importance of variables in predicting a response variable (NPP). The team selected a random forest model, using Burke et al. 2021 as proof-of-concept for applications in ecohydrology. Subsequent additions to the project included an additional workflow using gradient boosting to provide a comparison to random forest, as well as applying each workflow to additional RHESSys output variables. Further, the Sagehen Creek case study dataset had many derived variables that were incorporated into the analysis, such as uncertainty and climate scenarios, that are not directly output from the RHESSys model. Therefore, both random forest and gradient boosting workflows were duplicated and restructured in order to be more easily applied to standard RHESSys output files.

A data preparation workflow was created in order to streamline RHESSys users' data wrangling process prior to applying machine learning algorithms to the data. This workflow includes temporal aggregation, conversion of factor variables, creation of derived variables, and some unit conversion for ease of interpretation. Additionally, it creates a naming convention for a specific response variable in order to simplify the programmatic execution of several functions in subsequent workflows as well as the visualization app.

In order to assess variable importance, the first workflow uses Brieman's random forest technique—a machine learning method that builds an ensemble of decision trees (Brieman 2001). Each tree in the forest is grown using a bootstrapped dataset by sampling the original dataset with replacement. At each node, a number (chosen by the user) of explanatory variables are randomly chosen and assessed for their ability to explain the data via a decrease in sum of squared errors of the response variable. The algorithm iterates through this growing process and constructs a number of trees defined by the user—500 for the purposes of this project. Model evaluation is performed intrinsically as a part of the algorithm framework using out-of-bag samples. Random forests are able to make predictions and assess variable importance with a high degree of accuracy in numerous applications including ecological analysis (Cutler et al. 2007; Prasad et al. 2006).

Random forest variable importance can be measured with different methods, such as impurity and permutation variable importance. For this project, the team will focus on permutation importance, which involves shuffling or permuting a random predictor variable with each tree in a random forest to see the difference in prediction accuracy. Variables that have a larger difference on response prediction accuracy once permuted are considered more important. One limitation of permutation importance is that correlated variables create bias in results and erroneously lower variable importance (Strobl et al. 2008).

In order to resolve this limitation, the most highly correlated variables that affect importance and are removed prior to building the random forests. This is accomplished using variance inflation factors (VIF) and Pearson correlation thresholds. By default, variable preference order is created using importance from a preliminary random forest—however, users can manually enter preferred variable order. A high VIF value indicates that a variable is more significantly explained by all other predictor variables. A high magnitude Pearson correlation coefficient indicates that two variables are possibly measuring the same phenomenon through different mechanisms. The workflow uses default values of 5 for VIF and 0.75 for Pearson correlation, which can be altered by the user. This section will include a visual report of which features were selected and removed along with their associated correlation, VIF, and preliminary importance values.

In addition to random forest, the Capstone Team developed a gradient boosting workflow to provide another comparable method of gleaning variable importance. Similar to random forest, gradient boosting is a decision tree based method, however, gradient boosting uses a boosted ensemble technique in which models are iteratively trained to improve upon inaccuracies in previous models (Friedman 2002). Gradient boosting often outperforms random forest in terms of predictive capability, but the training process is more demanding in terms of time and computational resources; likewise, hyperparameter tuning is more complex and less approachable from the standpoint of a researcher broadly unfamiliar with machine learning (Freeman et al. 2016)—thus the Capstone Team decided to incorporate both methods. The gradient boosting workflow largely follows a similar trajectory as the random forest workflow, with the additional need to one-hot encode categorical variables.

Model accuracy (including inter-model performance) was evaluated to confirm the validity of both machine learning techniques, and the workflows were further applied to a new set of RHESSys output, which differed in exact variables, size, and use case, in order to evaluate the functionality of the workflow from an ecohydrology researcher's perspective. Improvements were made at various points in the workflows based on user feedback, including the separation of data preprocessing into a separate file, fully separating the random forest and gradient boosting workflows, and various modifications to the several functions the team developed to streamline the user interface with the workflows. Upon completion of this iterative evaluation process

(including user testing as outlined in the Testing section below), all machine learning workflows that the team developed were formalized and documented in order to serve as a framework for evaluating RHESSys output for future research. Using a combination of RMarkdown documents, R scripts, and accompanying material within the RHESSys GitHub wiki, the completed work is available through both GitHub and hydroshare.org to facilitate replication for new users and datasets.

Additionally, a Shiny interactive application was created for visualization of insights gained from machine learning processes. The basic properties of the Shiny app are interactive plots and tables that allow the user to select parameters such as climate scenario, topographic position, or response variable using sliders and drop-down menus. The resulting visualizations detail relevant findings related to the workflow, such as tables with rank order of predictor variable importance and plots with basic relationships between variables. These interactive visualizations are designed to give the user a more hands-on way to explore the relationships revealed by the variable importance derived from the machine learning models. This functionality is provided with the hope that users are much more comfortable creating and interacting with standard graphical representations of variable relationships; in this way, the Shiny app will provide a bridge between the somewhat esoteric world of machine learning and the tangible applications of the user's own research goals.

Software and Tools

This project has extensively used open source software and tools to analyze RHESSys output. The R language and RStudio open source environment were used to access, explore, interpret, and model RHESSys output. The renv package was used to ensure a consistent R environment in order to minimize potential errors caused by compatibility issues caused by package updates. The majority of R packages, functions, and related open source resources used for the machine learning component of this project were retrieved from CRAN.org. The exceptions include the ggbiplot package, which allows for plotting principal component analysis, and RHESSysIOinR. This package was specifically developed by RHESSys developers for running RHESSys and processing model output in an R environment, and can be accessed at https://github.com/RHESSys/RHESSysIOinR. Version control, timeline management, reproducibility, and public access for all stages of this project are available through GitHub at https://github.com/RHESSysML/RHESSysOutputExplorer.

PRODUCTS AND DELIVERABLES

There are three primary deliverables for this project:

1. Documented workflows to determine variable importance from RHESSys output, adaptable to any researcher's dataset.
2. Fully tested workflow implementation for RHESSys output data from Sagehen Creek.

3. Shiny application for data exploration and interactive visualization of workflow findings.

Both the documented workflows and the Shiny application are stored and accessible via GitHub, in addition to supplemental materials and explanatory documentation.

SUMMARY OF TESTING

Functionality Testing

Reports and visualizations are implemented at numerous stages of the process to serve as sanity checks against silent failures. Additionally, exceptions are included within the code that ensure user inputs are in the correct class, numeric range, or category list. These exceptions have been tested using the monkey testing approach, where various random inputs were input in an attempt to break the code.

Specific to the machine learning algorithms, different key parameter choices and resulting output are displayed in an effort to identify any potential silent failures or model biases as well as compare performance between models, though this may require substantial attention to detail or occasional "sanity checks" on behalf of the user. Ideally, RHESSys users, most of whom possess domain expertise in ecology and hydrology, will be able to visually identify inconsistencies.

User Interaction Testing

As reproducibility and re-usability by RHESSys users is an important objective of the project, user testing has been leveraged to identify and correct potential mistakes or areas of confusion.

The Tague Team Lab Manager, Janet Choate, was instrumental in providing feedback in the user testing process to ensure an easy to follow workflow, thorough documentation, and reproducible code. Janet is responsible for training and assisting new RHESSys users and delivered critical insight into typical user knowledge, familiarity with R and machine learning, and workflow usability. During the group's user testing meetings, certain pain points were identified and corrected in the final product. Data preparation, which encompasses all steps between creating RHESSys model output and running the machine learning workflow, was identified as the major hurdle. This resulted in this step being separated from the main workflow and receiving significant additional documentation and explanation. Additionally, locally created functions were moved to separate R scripts in order to condense the workflow.

Subsequently, the Capstone Team also requested user experience feedback from members of the Tague Team Lab, Louis Graup and Rachel Torres. These users were instrumental in providing feedback on the logical processes and statistical methods used in the workflow. During this stage

of user testing, it was acknowledged that certain statistical terms, such as out-of-bag sampling and split-improvement measures, needed further explanation. The workflow was updated to include reference papers on all relevant topics.

Shiny App Testing

The Shiny application deliverable required separate testing protocols from the machine learning and data visualization components of this Capstone Project. A monkey testing approach was used to ensure that no standard user input can break the application. This includes using reactive "observe" expressions that update user inputs based on other selections. Next, user testing was leveraged to improve the user experience and test usability by non-expert users. Dr. Tague assisted with ideas for exploratory visualizations, such as scatter plots showing the relationship between an independent and dependent variable, faceted by a third variable. Additionally, Rachel Torres provided feedback on the process of manipulating the application to work with a completely new dataset. This resulted in the Capstone Team creating two versions of the application, which will be discussed further in the "User Documentation" section below. Examples of changes to the Shiny app that came out of this user testing process include revisions to the introduction page and the addition of a metadata tab.

USER DOCUMENTATION

The purpose of this user documentation is to ensure a successful transition of this project from the Capstone Team to the Client - Dr. Naomi Tague and the Tague Team Lab. This has included transferring ownership of the GitHub repository and associated code, thorough explanation of the reasoning behind key decisions, and how to properly use and maintain the necessary tools.

How to access repository (GitHub ownership)

The Capstone Team's code and documentation is version controlled and saved on GitHub at the following organizational link - https://github.com/RHESSysML. The client and faculty advisor, Dr. Naomi Tague, has been given ownership of this organization to allow for the easy transition and integration of the Capstone Project into the Tague Team Lab's existing repositories.

The repository within the `RHESSysML` GitHub organization that contains the Capstone Team's work is titled `RHESSysOutputExplorer`. There are several folders and files within the repository, which are outlined below.

Repository structure

- `R/` - contains R functions used in the workflow and shiny application
- `data/` - contains all data used in the workflow, including the existing workflow implementations.
    - `input/` - contains original RHESSys output dataset and data files resulting from the data preparation notebooks.
    - `output/` - contains data files with model output from workflow notebooks.
    - `supporting_docs_data/` - contains data files used in the supporting docs found in `notebooks/supporting_docs`.
- `notebook_templates/` - contains template workflow notebooks for use with new datasets.
- `notebooks/` - contains completed notebooks used for the Team's workflow implementations.
    - `supporting_docs/` - contains notebooks supporting choices made in the workflow.
- `docs/` - folder to save RandomForestExplainer HTML output, if specified. Only applicable to the workflow using random forest.
- `shiny_sagehen/` - contains files and subdirectories associated with the `Shiny` interactive visualization application. This is the application for the sagehen creek dataset and other similar datasets (similar columns).
- `shiny_chap/` - contains files and subdirectories associated with the `shiny_chap` interactive visualization application. This is the application that is run with more general datasets.
- `renv/` - contains files and subdirectories created by the `renv` package (more information in part II of the section below)

How to use workflow

1. *Cloning the repository*

The workflow is intended to be used by forking and cloning the entire repository, then opening as an R Project in RStudio. Notebooks will not run outside of the project in their current state due to relative file paths and sourced functions.

2. *Using `renv` to limit compatibility issues*

The `renv` package was used to capture the environment in which the workflow was created—i.e. the RStudio version, packages, and package versions that were used to build it. As described in the workflow, users can install all necessary packages of the appropriate versions simply by

running `renv::restore()`, which the user will likely be prompted to do when the project is opened in RStudio. The resulting changes to package versions *only* occur within the project, and will not impact global package versions. The function only needs to be called once—the appropriate package versions should persist after the project is closed and reopened.

If it becomes necessary to update any aspects of the saved environment: install the necessary updates within the project and then call `renv::snapshot()` to create a new instance of the saved environment that contains those updates.

3. *Data preparation*

RHESSys output first needs to be prepared in a certain way. Users should start the workflow by using the `data_preparation_template.Rmd`. In this stage, variables are converted to their proper data type, the response variable is designated, and daily values of numeric variables are aggregated. It will require the user to actively choose the response variable and which variables should be converted to factors—places where user input is required are clearly denoted with commented text. The dataframes created are saved into a `prepared_data.RData` file which is next loaded into the `variable_importance_template.Rmd`.

4. *Variable importance*

Once the data are prepared and saved as `prepared_data.RData`, users should work through either the `rf_variable_importance_template.Rmd` or `gb_variable_importance_template.Rmd`, depending on which machine learning technique the user wants to use. Most of this process is fully automated, however, there are three important steps that require user attention—

- **Summarize data**: users should gut check the summary statistics from the prepared dataset to help ensure nothing unexpected happened in the data preparation process.
- **Multicollinearity removal**: users have the option to set a variable preference order manually in the multicollinearity removal step. By default, a preference order is set based on a preliminary assessment of variable importance—if users have specific variables they would like to be included in the model they can assign them to a preference order object manually.
- **Model evaluation**: model efficacy is assessed using a pseudo R^2 metric and is presented in the workflow as "percent variance explained." It is at the user's discretion to decide whether the model in their specific case is successful enough to deem the variable importance results useful. For reference, the Team's worked examples yielded models with roughly 90% variance explained.

5. *Shiny application*

The R Shiny application can be launched by opening any of the server.R, global.R, or ui.R files in the shiny application's directory and clicking "Run App" within R Studio. Additionally, the app can be run in a chunk at the bottom of the variable importance workflow. This will launch the application in a local browser window.

There are currently two shiny application versions in the repository that are designed around two workflow implementations. The first, which can be viewed in the `shiny_sagehen` folder, can be referenced to create an application to compare two scenarios. The second, `shiny_chap`, can be referenced to create an application that analyzes a single RHESSys simulation. It is important to note that the application relies on object names derived from the previous data preparation and variable importance steps—differences in specific datasets may require tweaking of the underlying code. For example, splitting the dataset for more than two scenarios.

In order for the R Shiny application to display the "Metadata" tab, it sources a file called `metadata.RDS`. This file is created in a file called `metadata.Rmd`, which is found in the respective shiny application folder. Currently, this file generates a metadata table for the Sagehen Creek dataset in both workflow implementations. Two functions are provided that allow the user to add or remove variables from the table. The table provides each variable's name as displayed in the data, full name, units, and a description. The file then exports the table as a `.RDS` file to be used in the Shiny app. This can be used as a template to include metadata for future applications.

Explanation of workflow implementation - Sagehen Creek

A full workflow example is supplied for RHESSys model output from the Sagehen Creek Experimental Watershed in the Sierra Nevada, CA. The associated files for steps 3-5 listed above are found within the repository at the following directories:

- (Data preparation): `notebooks/data_preparation_sagehen.Rmd`
- (Variable importance):
    - `notebooks/rf_variable_importance_sagehen.Rmd`; or
    - `notebooks/gb_variable_importance_sagehen.Rmd`
- (Shiny application): `shiny_sagehen/`

The data set incorporates unique variables for model parameter uncertainty (`scen`) and topographic variability under two separate climate warming scenarios (`clim`): (1) Historic temperature levels, and (2) Two degrees Celsius warming. This was accomplished by running multiple RHESSys simulations and binding the results into a single dataset, with unique combinations of `clim` and `scen` representing the simulations.

The Capstone Team chose to separate the Sagehen Creek case study data by these climate scenarios. Specifically, the team split the dataset into two separate dataframes, one for each climate scenario. This allows for analysis on the impact of climate on variable importance by calculating different importance ranks for each climate scenario, and comparing how each variable gains or loses importance in a warming climate.

The Capstone Team chose to incorporate this split to demonstrate interesting observations that can be made with unique data preparation steps. This example can be applied to datasets that contain a factor variable with two scenarios that the user would like to compare. This framework is not directly extensible for research projects that involve splitting the dataset into more than two scenarios.

The code for the Sagehen Creek workflow implementation is written with net primary productivity (NPP) as the response variable of interest. This means that the output of the example will offer an answer to the question: what are the most important ecohydrologic factors that affect NPP in an ecosystem like Sagehen Creek, and how might relative importance change in a warming climate?

Explanation of workflow implementation - Chap

Through user testing described above, the Capstone Team identified that many RHESSys users will run the workflow without splitting their dataset. In this case, all potential factor variables, such as “clim”, may be incorporated into a single model.

A second full workflow example is supplied for RHESSys model output for a chaparral ecosystem. This dataset entails RHESSys output at the Basin Daily Output level. The dataset was provided to the Capstone Team by Janet Choate. The associated files for steps 3-5 listed above are found within the repository at the following directories:

- (Data preparation): `notebooks/data_preparation_chap.Rmd`
- (Variable importance):
    - `notebooks/rf_variable_importance_chap.Rmd`; or
    - `notebooks/gb_variable_importance_chap.Rmd`
- (Shiny application): `shiny_chap/`

The dataset consists of a single RHESSys simulation. For users with a single RHESSys dataset, this implementation provides a concise framework to run the workflow.

Support for key decisions within workflow

1. *Removing multicollinearity*

For the purpose of assessing relative predictor variable importance using random forest, multicollinear variables have biased importance (Strobl et al. 2008). Therefore, highly correlated variables need to be handled prior to assessing variable importance.

2. *Random forest vs. gradient boosting*

The Capstone Team uses random forest as the primary method because it has been shown to be an effective tool in assessing variable importance in numerous applications, including ecological analysis (Cutler et al. 2007; Prasad et al. 2006). Additionally, random forest requires less hyper-parameter tuning than other common techniques.

The Capstone Team chose gradient boosting as a good alternative machine learning technique because it also uses a tree-based modeling approach. Gradient boosted models can often attain greater predictive accuracy than random forests, albeit at the cost of a more nuanced tuning process and a computationally intensive training process. The introduction to the gradient boosting variable importance file provides further detail regarding the tradeoffs between random forest and gradient boosting, and provides suggestions on why a user might choose to use one or the other.

3. *Decisions addressed in supporting documents*

In an effort to keep the primary workflow concise while elucidating key decisions and assumptions, the Capstone Team developed a series of supporting documents. These files can be found in the `notebooks/supporting_docs/` folder and are explained in detail below.

`Data_preparation_categoricals.Rmd` and `rf_variable_importance_cats.Rmd` were created to demonstrate running the workflow on only a subset of the predictor variables. Specifically, only non-numeric variables were modeled, which for the Sagehen Creek data included `stratumID`, `clim`, and `scen`. It is important to note that there are other potential static variables that may be of interest in this dataset. For instance, each `stratumID` has an associated value for elevation, slope, and aspect. However, decision tree models such as random forest and gradient boosting rely on variation in numeric variables to accurately predict a response. Since NPP varies while these variables remain static, they have limited predictive power. These are also confounding variables with `stratumID`.

Additionally, the `rf_variable_importance_cats.Rmd` file examines the difference between using the `randomForest` implementation of categorical variables, which allows for directly modeling these variables without transformation, by using one-hot encoding (OHE). OHE is a common method used in machine learning techniques to transform categorical variables into a numeric, machine readable format. This is required for gradient boosting as well as implementations of random forest in other programming languages, such as Python. The resulting variable importance and model fit comparisons indicate that there is not a significant difference between the two methods. Since decision trees benefit from reduced dimensionality, the primary workflow does not use one-hot encoding.

The, `partial_vs_conditional_importance.Rmd` tests the conditional permutation importance (CPI) implementation for random forest models from the `permimp` package. This method was developed by Strobl and Debeer (2020). CPI has been shown to mitigate some of the bias incurred by highly collinear predictor variables when using partial permutation importance (Strobl et al. 2008). The Strobl and Debeer paper also tested the impacts of numeric variables with varying range and categorical variables with different numbers of groups, both of which are common characteristics of RHESSys data. The test document did not reveal meaningful variable importance results using CPI. Variable importance was concentrated on very few variables, which does not inform exploratory data analysis as intended by the Capstone Project. In addition to these results, the Capstone Team proceeded with partial permutation importance because it is a more widely used method.

The `principal_component_analysis.Rmd` was used to explore using principal component analysis as another exploratory data analysis tool to accompany the workflow. Ultimately, the code and results from this file were incorporated into the “Principal Component Analysis” tab of the shiny application.

The `random_seed_comparison.Rmd` was used to test the random forest workflow results using numerous random seeds. Random forest models differ between programming language implementations, seeds, etc. due to the underlying random number generation when performing bootstrap sampling. The effect of this randomness is introduced for the preliminary importance model used to determine preference order, during tuning of the `mtry` hyper-parameter, as well as in the model used to generate variable importance. This file tested five versions of the final models, only differing by seed. The results indicate that variable importance values are relatively consistent for random seeds. However, in cases where two variables have nearly identical importance values, it is possible that their relative rank will change.

The `testing_vif_functions.Rmd` primarily tests the different methods for creating a preference order for use in the `remove_vif()` function. The options are 1) using a preliminary random forest model to determine preference order based on variable importance, 2) setting

preference order equal to `NULL`, which sets preference order based on VIF value, and 3) determining a preference order manually. Additionally, these results were compared with those from a different function, `vif_remove()`, which sequentially removes the highest VIF values until all variables are within a specified threshold. The results are provided to assist with the decision to choose a method of using VIF to remove highly collinear variables.

Limitations

1. *Interpretation of variable importance*

As discussed within the respective workflow notebooks, variable importance derived from machine learning models has some interpretability limitations. "Importance" can be determined in a variety of manners, all of which relate to determining which variables are most helpful in improving a model's predictive power.

In particular, permutation importance is linked to the error of the model. As described previously, this metric measures the decrease in model performance when permuting a variable. For certain particular research questions, other measures of performance may be more directly applicable. An example would be testing the robustness of the random forest model's output to an artificially manipulated variable. In this case, it may be more relevant to measure importance as the percent variance explained by that feature (Molnar, 2022).

Despite these limitations, permutation importance can be easily interpretable, provides an overall insight into a dataset and model, and values are easily comparable. Additionally, permutation importance accounts for variable interactions that are not accounted for by linear models. Other importance metrics for random forest are available through the RandomForestExplainer package that can be optionally used in the workflow.

2. *"Black box" of machine learning models*

Understanding what's happening "under the hood" in machine learning models can be difficult, especially when compared to process-based models such as RHESSys. For users unfamiliar with machine learning, some of the more specific fine-tuning processes within the workflow may be daunting. Furthermore, a lack of immediate, straightforward interpretability of predictive models may impact researcher's trust in the results. The Shiny application was built to tie output from the random forest and gradient boosting models to commonly used exploratory data analysis, such as time series, distributions, and scatter plots, in an effort to reduce this information gap.

Future directions - within the workflow's scope

1. *Generalizing workflow and shiny application*

Currently, there are differences between the two workflow implementation examples, as explained above. This inflexibility means that users may have issues implementing the workflow for different cases, such as comparing more than two scenarios. Additionally, certain areas of the workflow expect consistent columns, such as `wy` or "water year". Overall, the Capstone Team expects a relatively easy transition of the variable importance files, while the shiny application may be more difficult to manipulate.

One potential idea that may solve this is utilizing list objects with nested data frames. This would allow the same action to be performed for each data frame, regardless of the number of scenarios. For instance, the Sagehen Creek workflow implementation splits the data into two scenarios using the `dplyr::filter()` function. Instead the `base::split()` function could be used, which creates a list of dataframes based on a specified group. The model output from the variable importance files would also need to be placed in list objects. This could potentially make the shiny application more flexible to different use cases.

2. *Temporal aggregation*

One of the Capstone Team's most difficult decisions was how to determine the best temporal aggregation of daily RHESSys output data for use with the machine learning models. The team chose to aggregate data by water year while also creating variables relevant to this time scale, including peak annual snow water equivalent (SWE) and seasonal temperature. Future researchers, who may want to analyze data at a different temporal aggregation, will have to consider the statistical implications of this change throughout the workflow. Thus, there is an opportunity to improve upon this workflow by providing more analysis and research into how to best aggregate data for this purpose.

3. *Other measures of variable importance*

In this workflow, the Capstone Team relies on tree-based models for determining variable importance. However, there are numerous other ways of assessing variable importance, such as through the use of support vector machines or linear regression. While random forest models have been thoroughly used for this purpose in many fields including ecology, another method of determination may prove to be more useful for some specific research questions.

Future directions - outside of the workflow's scope

1. *Time-Series Analysis*

The Capstone Team began exploring ways to incorporate time series analysis into the project's scope. However, this would serve a separate purpose from the Project's main objective of assessing variable importance. One such analysis that the Capstone Team began is identifying memory effects using recurrent neural networks (RNN), specifically long short-term memory (LSTM) architectures. For further reading, please see Kraft et al. 2019 and the `RHESSysOutputLSTM` repository.

Data & metadata

1. *Workflow Implementation Datasets*

The Capstone Team formulated the primary workflow around RHESSys model output for the Sagehen Creek Experimental Watershed in the Sierra Nevada, CA. The dataset incorporates model parameter uncertainty, topographic spatial variability, and climate change effects.

The second workflow implementation is basin level daily output for a chaparral ecosystem provided by the Tague Team Lab. For metadata or further information about the characteristics of this dataset, please contact Janet Choate.

2. *Uniqueness of Sagehen Creek Dataset*

The Sagehen Creek dataset contains non-standard RHESSys output columns, including the `topo`, `clim`, `scen`, and `wy` variables. The `topo` variable contains categories that represent different topographic positions within a watershed. The `clim` variable contains categories that represent the climate scenario used in RHESSys simulations - either normal temperature or a plus two degree celsius warming scenario. The `scen` variable contains categories that represent different input parameter sets used in RHESSys simulations. The `wy` variable represents water year and is derived from the `day`, `month`, and `year` columns.

Additionally, the Capstone Team investigated certain variables derived from standard RHESSys output, such as `peak_swe` and `swe_precip_ratio`. Information on how these variables were created can be found within the workflow. The remaining variables within the Sagehen Creek dataset are standard RHESSys output data that can be expected from a simulation at stratum daily output resolution.

ARCHIVE ACCESS

Sagehen Creek dataset specifications

Louis Graup of the Tague Team Lab generated the Sagehen dataset using RHESSys on January 10, 2022. The file is 220MB uncompressed and contains 1,183,380 daily observations among 30 variables in CSV format.

The dataset and accompanying metadata are currently accessible on Hydroshare: https://www.hydroshare.org/resource/2a31bd57b7e74c758b7857679ffbb4c5/. The specific RHESSys input parameters and preparatory code to prepare simulation output are available upon request via the client.

Metadata

There is currently metadata on the dataset's hydroshare.org page that includes a brief description of the dataset, source information, study site information, and variable descriptions. Additionally, the Capstone Team has included more detailed metadata for the Sagehen dataset in the Shiny application.

Data sharing and access

Data from this project will be available online for others to use and access without any significant limitations. Data can be accessed either through hydroshare.org or the GitHub repository.

Data archiving and preservation

Raw data is preserved in CSV format on hydroshare.org, where it can be accessed along with accompanying metadata. The code for the workflow and Shiny application is stored and accessible via the RHESSysOutputExplorer GitHub repository at https://github.com/RHESSysML/RHESSysOutputExplorer.

REFERENCES

Breiman, Leo. 2001. "Random Forests." *Machine Learning* 45 (1): 5–32. https://doi.org/10.1023/A:1010933404324.

Burke, William D., Christina Tague, Maureen C. Kennedy, and Max A. Moritz. 2021. "Understanding How Fuel Treatments Interact With Climate and Biophysical Setting to Affect Fire, Water, and Forest Health: A Process-Based Modeling Approach." *Frontiers in Forests and Global Change* 3: 143. https://doi.org/10.3389/ffgc.2020.591162.

Cutler, D. Richard, Thomas C. Edwards, Karen H. Beard, Adele Cutler, Kyle T. Hess, Jacob Gibson, and Joshua J. Lawler. 2007. "Random Forests for Classification in Ecology." Ecology 88 (11): 2783–92. https://doi.org/10.1890/07-0539.1.

Debeer, Dries, and Carolin Strobl. 2020. "Conditional Permutation Importance Revisited." *BMC Bioinformatics* 21 (1): 307. https://doi.org/10.1186/s12859-020-03622-2.

Freeman, Elizabeth A., Gretchen G. Moisen, John W. Coulston, and Barry T. Wilson. 2016. "Random Forests and Stochastic Gradient Boosting for Predicting Tree Canopy Cover: Comparing Tuning Processes and Model Performance." *Canadian Journal of Forest Research* 46 (3): 323–39. https://doi.org/10.1139/cjfr-2014-0562.

Friedman, Jerome H. 2002. "Stochastic Gradient Boosting." *Computational Statistics & Data Analysis*, Nonlinear Methods and Data Mining, 38 (4): 367–78. https://doi.org/10.1016/S0167-9473(01)00065-2.

Guswa, Andrew J., Doerthe Tetzlaff, John S. Selker, Darryl E. Carlyle-Moses, Elizabeth W. Boyer, Michael Bruen, Carles Cayuela, et al. 2020. "Advancing Ecohydrology in the 21st Century: A Convergence of Opportunities." *Ecohydrology* 13 (4): e2208. https://doi.org/10.1002/eco.2208.

Kraft, Basil, Martin Jung, Marco Körner, Christian Requena Mesa, José Cortés, and Markus Reichstein. 2019. "Identifying Dynamic Memory Effects on Vegetation State Using Recurrent Neural Networks." *Frontiers in Big Data* 2. https://www.frontiersin.org/article/10.3389/fdata.2019.00031.

Molnar, C. (2022, March 29). Interpretable machine learning. 8.5 Permutation Feature Importance. Retrieved June 3, 2022, from https://christophm.github.io/interpretable-ml-book/feature-importance.html#disadvantages-9

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): 181–99. https://doi.org/10.1007/s10021-005-0054-1.

Strobl, Carolin, Anne-Laure Boulesteix, Thomas Kneib, Thomas Augustin, and Achim Zeileis. 2008. "Conditional Variable Importance for Random Forests." *BMC Bioinformatics* 9 (1): 307. https://doi.org/10.1186/1471-2105-9-307.

Tague, C. L., and Band, L. E. “RHESSys: Regional Hydro-Ecologic Simulation System—An Object-Oriented Approach to Spatially Distributed Modeling of Carbon, Water, and Nutrient Cycling.” *Earth Interactions* 8, no. 19 (December 1, 2004): 1–42. https://doi.org/10.1175/1087-3562(2004)8<1:RRHSSO>2.0.CO;2.

Tague, C., & Frew, J. (2021). Visualization and ecohydrologic models: Opening the box. *Hydrological Processes*, *35*(1), e13991. https://doi.org/10.1002/hyp.13991

Tague, Christina, and Hui Peng. “The Sensitivity of Forest Water Use to the Timing of Precipitation and Snowmelt Recharge in the California Sierra: Implications for a Warming Climate.” *Journal of Geophysical Research: Biogeosciences* 118, no. 2 (2013): 875–87. https://doi.org/10.1002/jgrg.20073.

Thessen, Anne. 2016. “Adoption of Machine Learning Techniques in Ecology and Earth Science.” One Ecosystem 1 (June): e8621. https://doi.org/10.3897/oneeco.1.e8621.