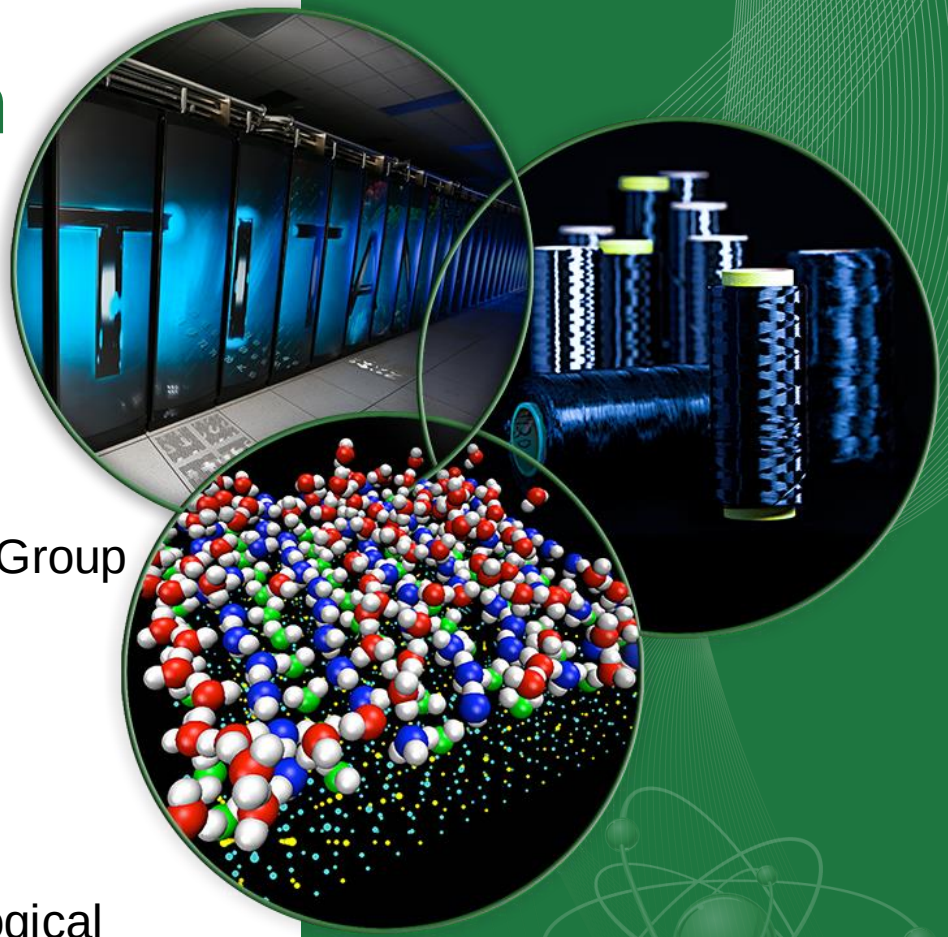
User-friendly data-driven machine learning for simulating vegetation dynamics



Elias Massoud*, Morgan Steckler,
Siyan Liu, Nathan Collier,
Jitendra Kumar, Forrest Hoffman

Integrated Computational Earth Sciences Group
Oak Ridge National Laboratory

American Meteorological Society 2025
Wednesday January 15, 2024

Session: Data-Driven Methods for Hydrological Modeling, Prediction, and Uncertainty Estimation
Conference: 39th Conference on Hydrology

ORNL is managed by UT-Battelle
for the US Department of Energy

RUBISCO

OAK RIDGE National Laboratory

Outline

- **Synthesis of Leaf-Level Photosynthesis Data**:
  - Data sources and preparation (creating an “AI-ready” dataset)
  - Synthesis workflow: Steps to prepare ML-ready datasets
  - Processing example: Lin 2015 dataset demonstration
- **User-friendly Machine Learning**:
  - Configure ML vegetation dynamics models
  - User configurations: Customizable options for ML simulations
- **Comparing Performance of ML Models**:
  - Visualization: Scatter plots of predictions vs. true values
  - Model performance (Training time, RMSE, R^2)

OAK RIDGE National Laboratory

Synthesis of Leaf-Level Photosynthesis Data

OAK RIDGE National Laboratory

“AI-ready” Photosynthesis dataset

Creating this AI-ready dataset entails:

- Leveraging curated datasets for leaf-level fluxes (e.g., for photosynthesis and stomatal conductance).

- Data processing: handling outliers, missing values, and standardization.

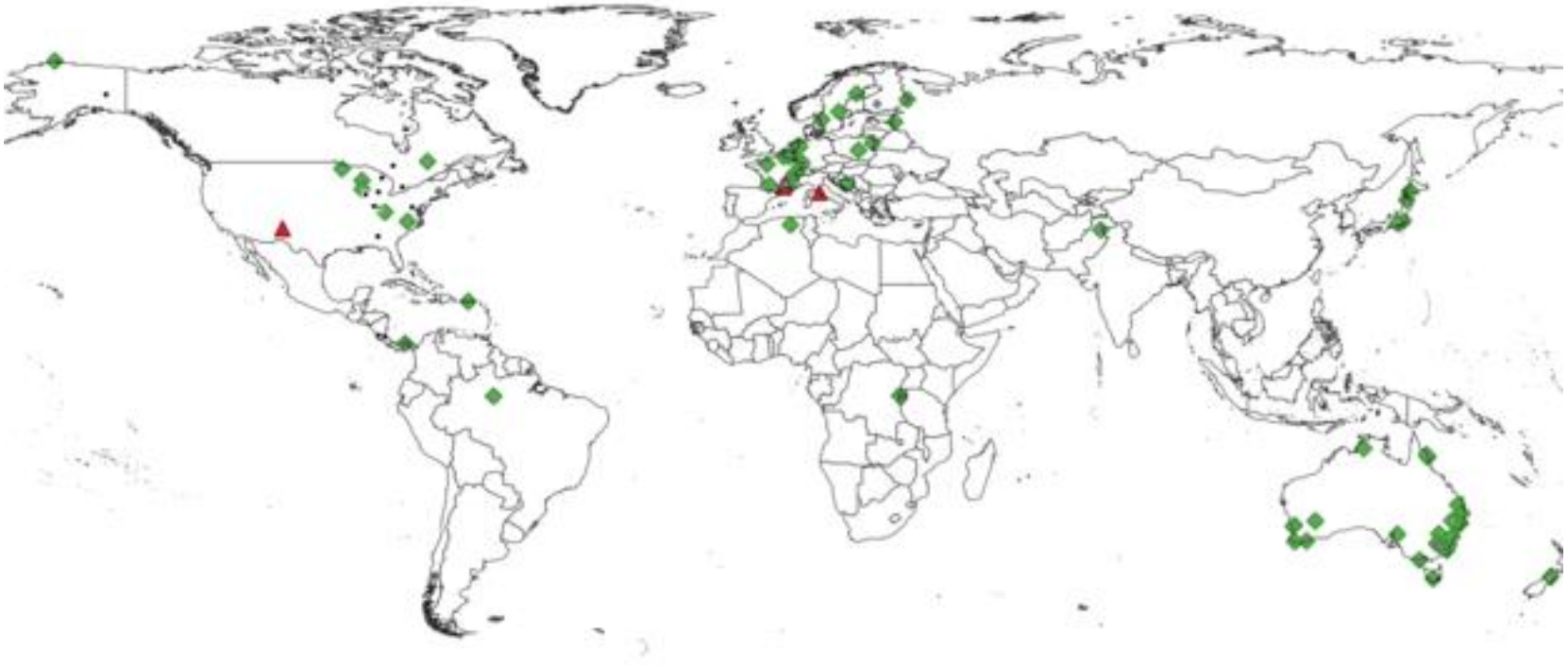
- Dataset is synthesized according to DOE’s ESS-Dive leaf-gas-exchange guidelines

- Collections include:
  - Lin 2015
  - Anderegg 2018
  - Saunders 2021
  - Han 2022
  - Various data in ESS-Dive and other leaf-level measurements

OAK RIDGE National Laboratory

Spatial Distribution of Data

”Global” spatial coverage of datasets,
… but still some gaps (e.g., tropics and global south)



This coverage still seems better than most available datasets!

Synthesis of leaf-level photosynthesis data

Data Variables		Metadata Variables	
Tleaf	Leaf surface temperature	**Species Information:**	**Source Information:**
A	Net CO2 exchange per leaf area	recordId	dataContributor
Ci	Intercellular CO2 concentration	unadjudicatedName	dataFile
CO2s	CO2 concentration in wet air inside	acceptedName	dataCitation
RHs	Relative humidity of air inside	nameAuthority	dataHtml
Tair	Air temperature inside	nameAuthor	
Qin	In-chamber photosynthetic flux density	nameCategory	**Spatiotemporal Information:**
Patm	Atmospheric pressure	adjudicator	siteIdentifier
gsw	Stomatal conductance to vapor	plantType	latitudeY
gbw	Boundary layer conductance to vapor	plantForm	longitudeX
E	Transpiration rate of H2O	plantLeaf	year
VPDleaf	Leaf to air vapor pressure deficit	plantHeight	month
area	Leaf area	canopyPosition	day
Tblock	Temperature of cooler block		time
CO2r	CO2 concentration in wet air entering	**Survey Information:**	season
RHr	Relative humidity of air entering	surveyType	country
Flow	Flow rate of air entering	measurementDevice	state
Qout	External photosynthetic flux density	growthEnvironment	plantBiome
CiCa	Ratio of intercellular CO2 to chamber CO2	experimentalManipulation	
LAI	Leaf area index		

OAK RIDGE National Laboratory

Synthesis Workflow

Step 1: Identify datasets with leaf-level data.

Step 2: Standardize variables across datasets (e.g., units, time steps).

Step 3: Integrate metadata (e.g., location, environmental factors).

Result: An AI-ready dataset suitable for ML models.

OAK RIDGE National Laboratory

Diagram of Synthesis Workflow

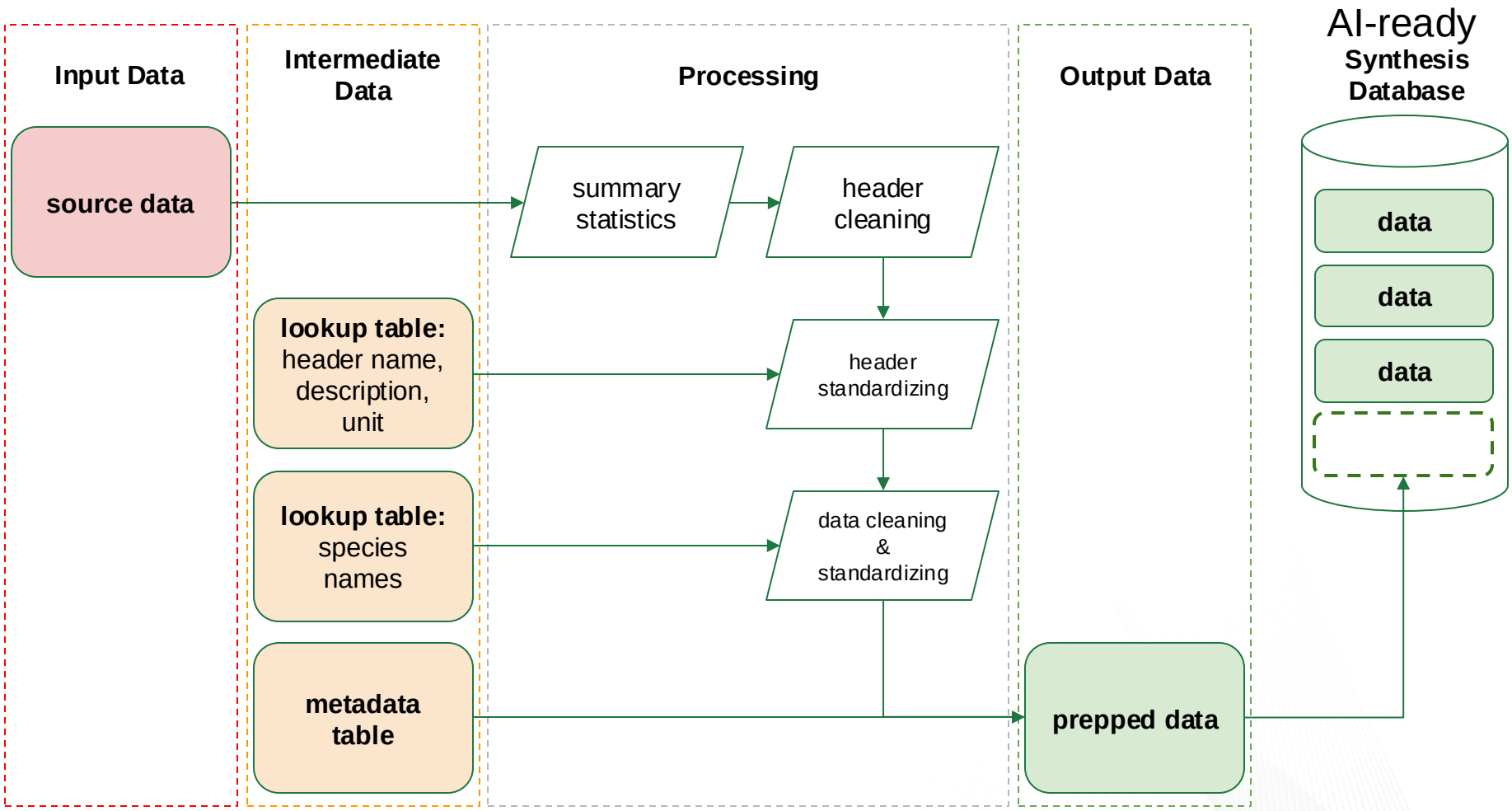


Figure credit: Morgan Steckler

OAK RIDGE National Laboratory

Lin 2015 processing example

Contains data from 44 contributors

Unique date strftime formats:

%d/%m/%Y
%Y%m%d
%b
%B
%b %d %Y
%d/%m/%y
%d-%b-%y
%m/%d/%Y
%Y
%d-%b
%Y-%m-%d
%d.%m.%Y
%d/%m/%Y %H:%M

Formats show there is missing year/month/day information from some contributors

Species names issues

- **Abbreviations:** 'euc saligna de' should be 'eucalyptus salinga'
- **Source UIDs:** to differentiate the data source, lin 2015 added the author initials at the end of the species name
  - E.g. 'fagus sylvatica mf', 'fagus sylvatica ml', 'fagus sylvatica rc'
  - Also 'repeat' added at the end of species names
- **Place UIDs**: Things like 'stpaul vic la gardiole' after a species name.
  - Remove these but ensure there is a siteIdentifier associated with the species
- **Common names**: like 'bindweed'
- **Genus or family names**: instead of a species name we might find 'cenchrus', 'alysicarpus', etc.
- **Misspellings:** ('caraipa' vs 'carapa')

OAK RIDGE National Laboratory

Lin 2015 processing example

Contains data from 44 contributors

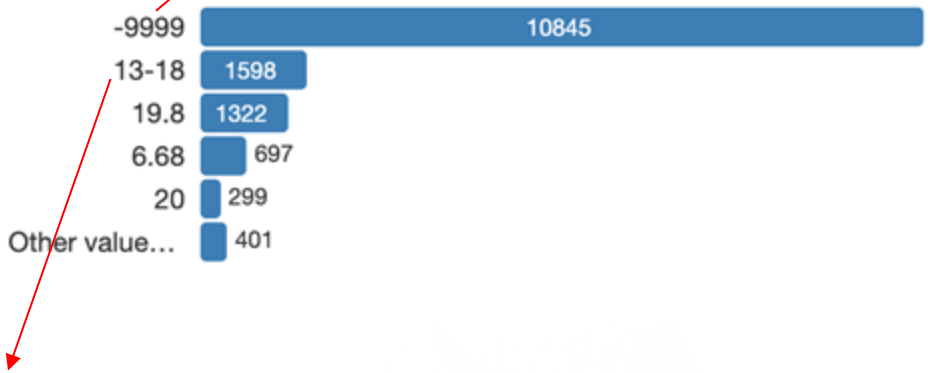
Totalheight
Categorical

Convert Totalheight to plantHeight; need to add description and determine units

HIGH CORRELATION IMBALANCE

Distinct	16
Distinct (%)	0.1%
Missing	0
Missing (%)	0.0%
Memory size	118.6 KiB

Convert no-data value to None; majority of plantHeight column contains no-data

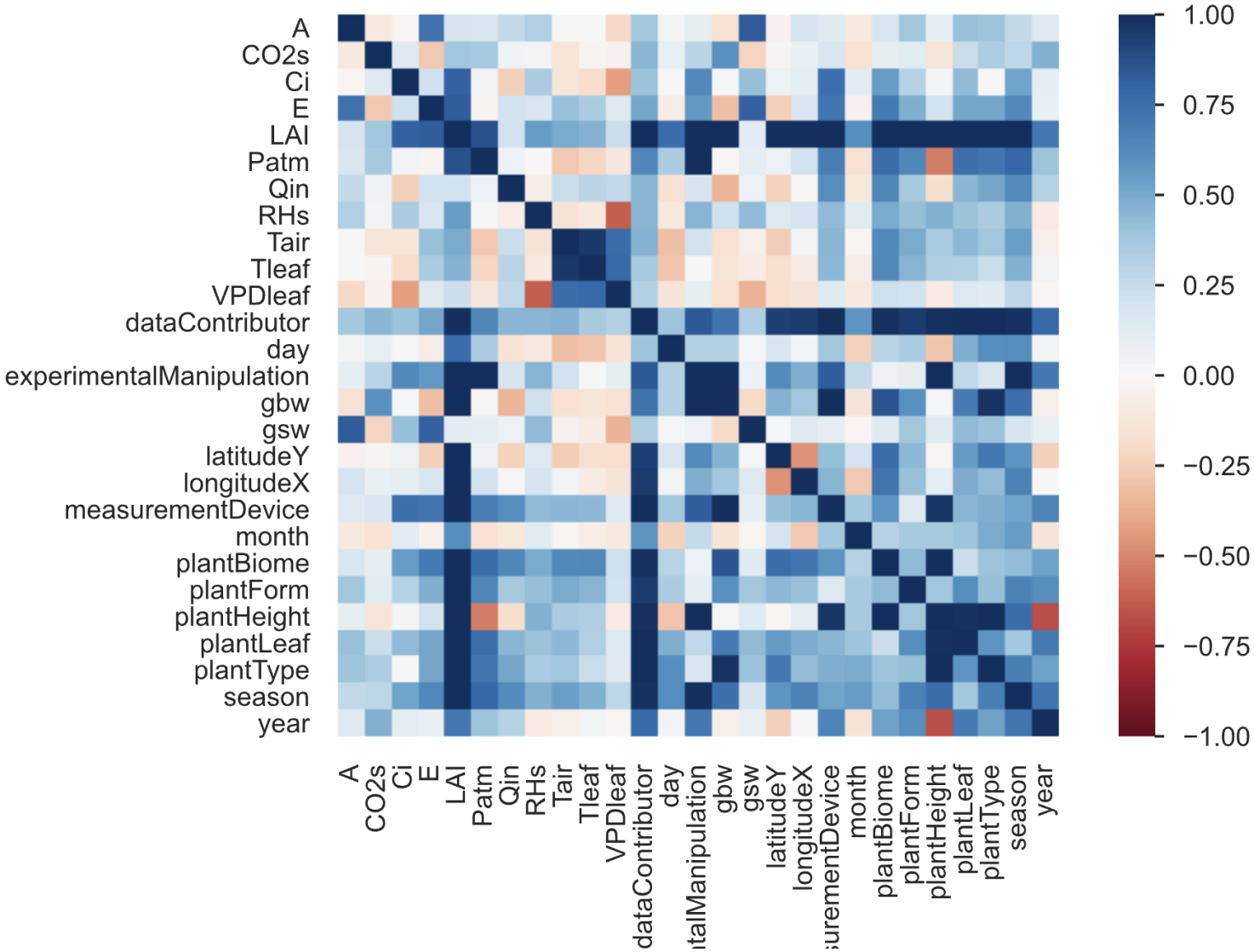


Mixed data types: strings, floats, ints

OAK RIDGE National Laboratory

Correlation Between Variables

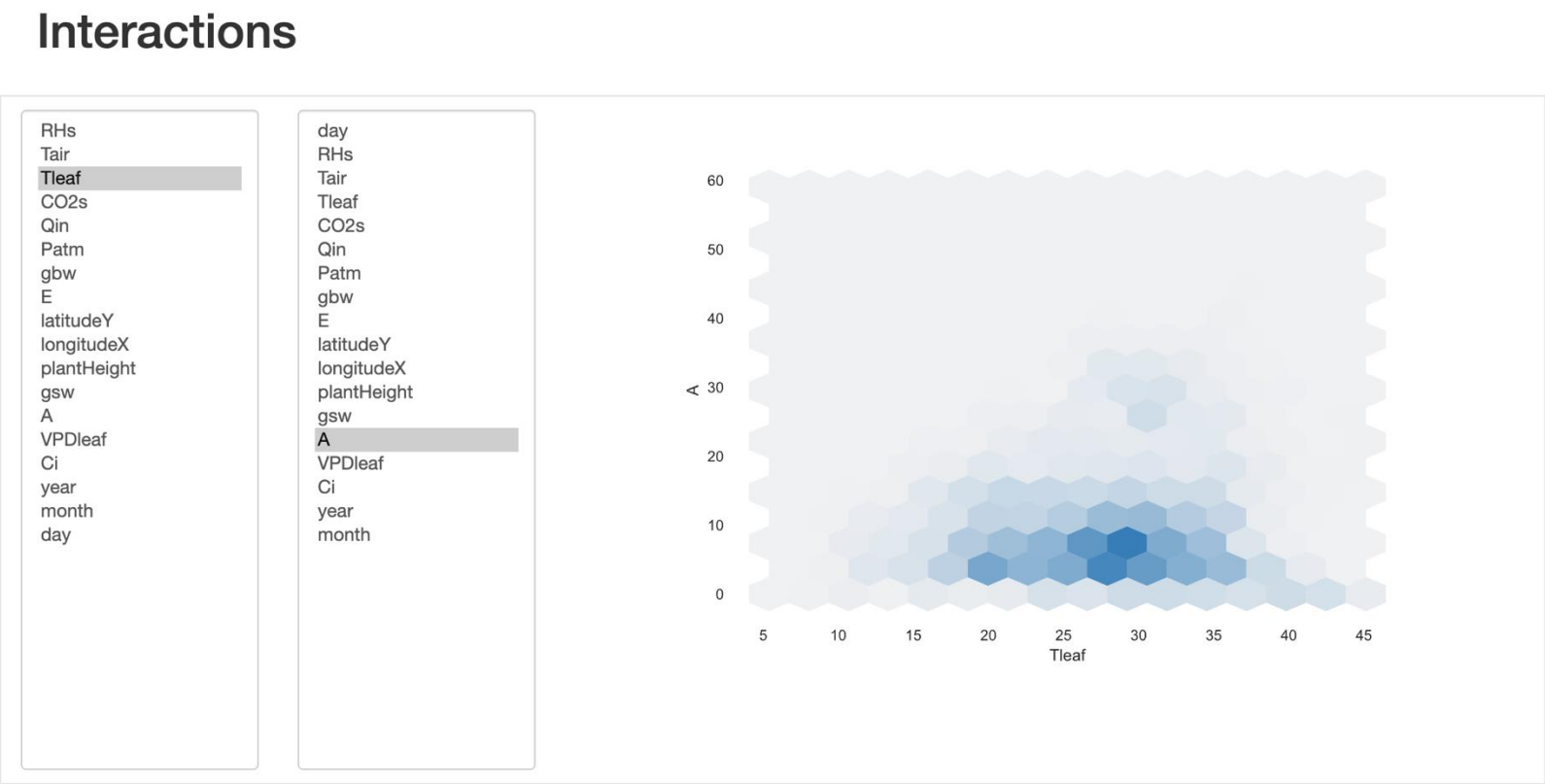
To first order, we can check correlation between variables



Correlation Between Variables

We can even check more detailed correlation between individual variables.

This example shows photosynthesis (A) with leaf temperature (Tleaf).



User-friendly Machine Learning

OAK RIDGE National Laboratory

Machine Learning Setup

These are the things involved in the Machine Learning Setup:

Dataset Selection: Choose different datasets or subsets.

Model Configuration: Customize layers, nodes, and activation functions.

- Model architecture: Configurable layers (e.g., 2–5 hidden layers)
- Nodes per layer (e.g., 50–200).
- Activation functions (ReLU, sigmoid).
- Predefined recipes for rapid testing and experimentation.
- Hyperparameter tuning: optimizer, learning rate, and batch size.
- Example ML algorithms used: MLPRegressor, NN's, Random Forest.

Training Settings: Adjust epochs, batch size, or optimizer.

Visualization Options: Select plots or metrics for evaluation.

OAK RIDGE National Laboratory

ML Setup

1. Build model

```python
def build_model(input_size, output_size, hidden_layers):
    """
    Build a neural network model based on the configuration.

    Parameters:
    - input_size: Size of the input layer.
    - output_size: Size of the output layer.
    - hidden_layers: List of hidden layer sizes.

    Returns:
    - model: The neural network model.
    """
    print('--- Input size:', input_size)
    print('--- Output size:', output_size)
    model_layers = [nn.Linear(input_size, hidden_layers[0]), nn.ReLU()]
    for i in range(len(hidden_layers) - 1):
        model_layers.append(nn.Linear(hidden_layers[i], hidden_layers[i + 1]))
        model_layers.append(nn.ReLU())
    model_layers.append(nn.Linear(hidden_layers[-1], output_size))
    model = nn.Sequential(*model_layers)
    return model
```

Figure credit: Siyan Liu

OAK RIDGE National Laboratory

ML Setup

2. Initialize model

```python
def initialize_model(input_size, output_size, hidden_layers, learning_rate):
    """
    Initialize the model, criterion, and optimizer.

    Parameters:
    - input_size: Size of the input layer.
    - output_size: Size of the output layer.
    - hidden_layers: List of hidden layer sizes.
    - learning_rate: Learning rate for the optimizer.

    Returns:
    - model: The neural network model.
    - criterion: The loss function.
    - optimizer: The optimizer.
    """
    model = build_model(input_size, output_size, hidden_layers)
    criterion = nn.MSELoss()  # Assuming regression task, adjust if classification
    optimizer = optim.Adam(model.parameters(), lr=learning_rate)
    return model, criterion, optimizer
```


OAK RIDGE National Laboratory

ML Setup

3. Training steps

```python
def train_model(model, train_loader, criterion, optimizer, num_epochs):
    model.train()
    for epoch in range(num_epochs):
        print(f'Epoch [{epoch+1}/{num_epochs}]')
        epoch_loss = 0.0
        with tqdm(total=len(train_loader), desc=f'Epoch {epoch+1}/{num_epochs}', unit='batch') as pbar:
            for inputs, labels in train_loader:
                optimizer.zero_grad()
                outputs = model(inputs)
                loss = criterion(outputs, labels)
                loss.backward()
                optimizer.step()
                epoch_loss += loss.item()
                pbar.set_postfix(loss=loss.item())
                pbar.update(1)
        print(f'Epoch [{epoch+1}/{num_epochs}], Loss: {epoch_loss/len(train_loader):.4f}')
    return model
```

4. Evaluation steps

```python
def evaluate_model(model, test_loader):
    model.eval()
    all_labels = []
    all_predictions = []
    with torch.no_grad():
        for inputs, labels in test_loader:
            outputs = model(inputs)
            all_labels.extend(labels.numpy())
            all_predictions.extend(outputs.numpy())

    # Calculate RMSE and R²
    rmse = mean_squared_error(all_labels, all_predictions, squared=False)
    r2 = r2_score(all_labels, all_predictions)
    return rmse, r2
```


OAK RIDGE National Laboratory

Where can we make changes/edits

Now that the data and the ML scripts are set up …
we can quickly do testing by making changes to:

Dataset Selection: Choose different datasets or subsets.

Model Configuration: Customize layers, nodes, and activation functions.

Training Settings: Adjust epochs, batch size, or optimizer.

Visualization Options: Select plots or metrics for evaluation.

Easy-to-use scripts enable quick testing and experimentation.

OAK RIDGE National Laboratory

Where people can make changes/edits

Use configure file to define the ML network structures, Hyper-parameters, etc.

models > saunders2021_SL > ! config.yaml

```yaml

data:
  catalog_path: "../../leaf-level.yaml"
model:
  layers:
    - hidden_size: 64
    - hidden_size: 32
    - hidden_size: 32
training:
  num_epochs: 600
  learning_rate: 0.001
  batch_size: 512
```


OAK RIDGE National Laboratory

Where people can make changes/edits

A few lines of code run the entire analysis!

```python
if __name__ == "__main__":
    config = load_config('config.yaml')

    hidden_layers = [layer['hidden_size'] for layer in config['model']['layers']]
    num_epochs = config['training']['num_epochs']
    learning_rate = config['training']['learning_rate']
    batch_size = config['training']['batch_size']
    catalog_path = config['data']['catalog_path']

    # Create DataLoaders and get input/output sizes
    train_loader, test_loader, input_size, output_size, x_test_tensor, y_test_tensor = create_dataloaders(catalog_path, batch_size)

    # Initialize model, criterion, and optimizer
    model, criterion, optimizer = initialize_model(input_size, output_size, hidden_layers, learning_rate)

    # Train and evaluate the model
    model = train_model(model, train_loader, criterion, optimizer, num_epochs)
    rmse, r2 = evaluate_model(model, test_loader)

    print(f'RMSE: {rmse:.4f}')
    print(f'R²: {r2:.4f}')
```


OAK RIDGE National Laboratory

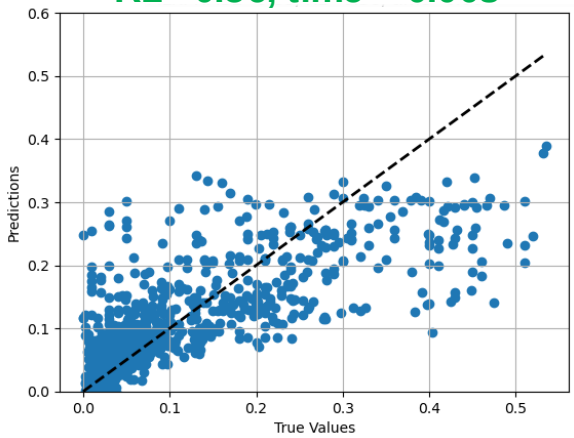
Comparing Performance of ML Models

OAK RIDGE National Laboratory

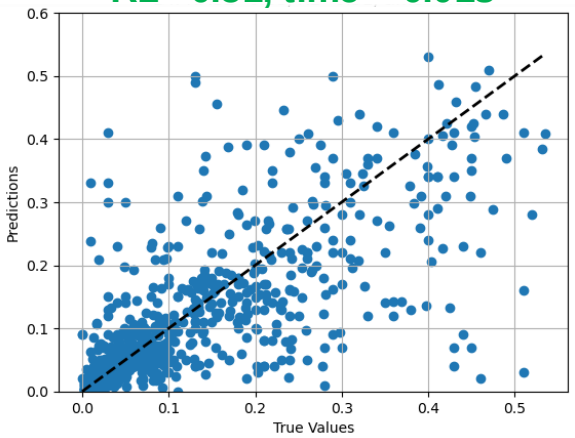
Prediction vs true value

Visual comparison of predictions and true values for key datasets

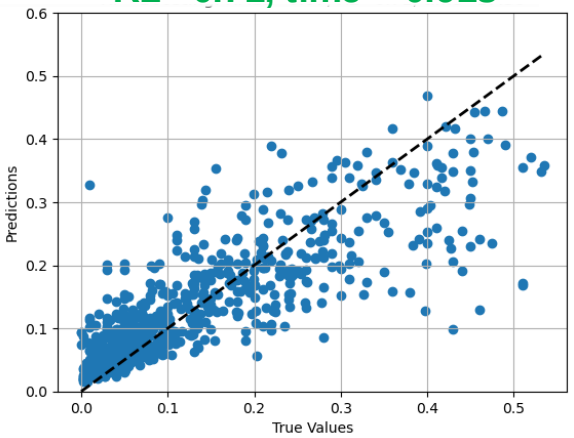
MLR: RMSE = 0.08, R2 =0.56, time = 0.00s



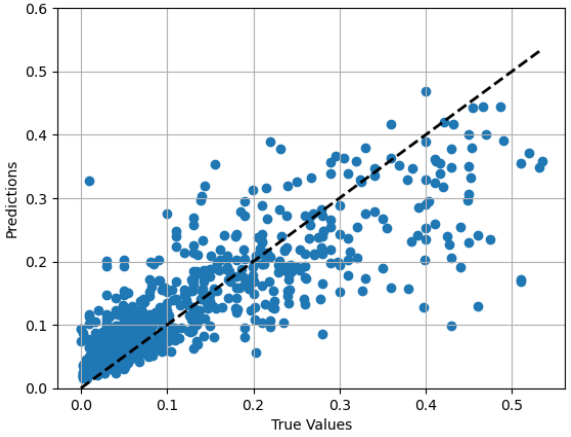
DT: RMSE = 0.08, R2 =0.51, time = 0.01s



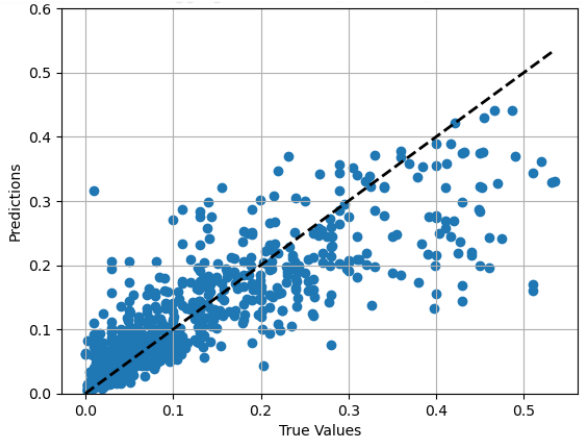
RF : RMSE = 0.07, R2 =0.71, time = 0.61s



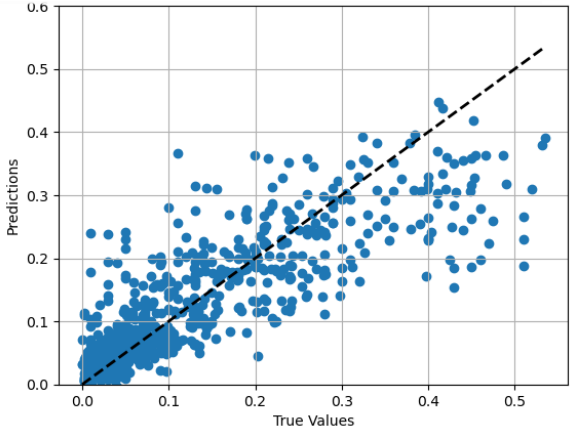
RF - Bagging: RMSE = 0.06, R2 =0.71, time = 3.72s



RF - Boosting: RMSE = 0.07, R2 =0.71, time = 17.54s



NN: RMSE = 0.06, R2 =0.71, time = 0.57s



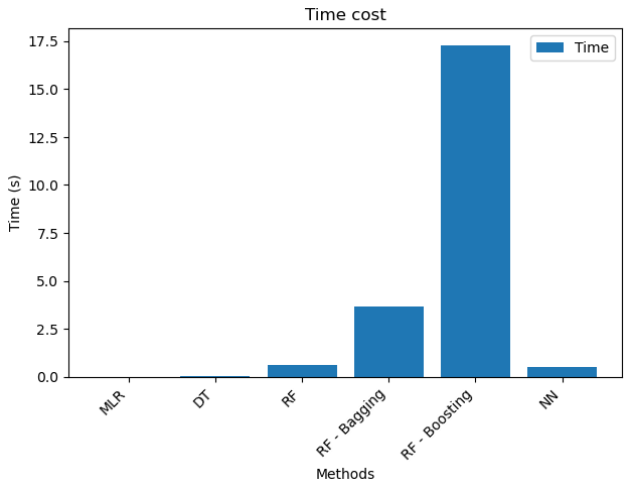
OAK RIDGE National Laboratory

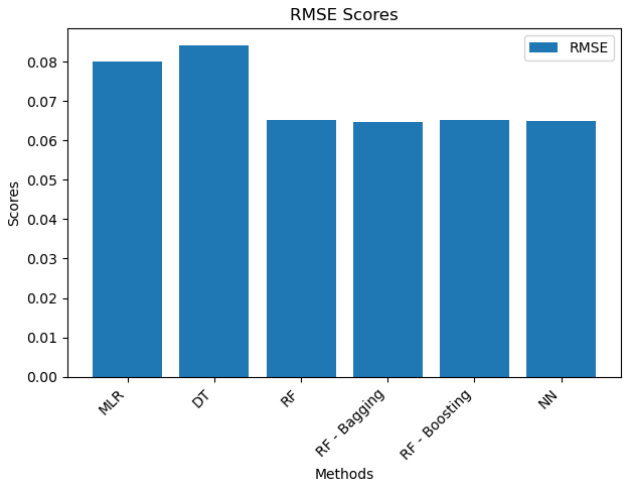
Training time, RMSE and R2 scores

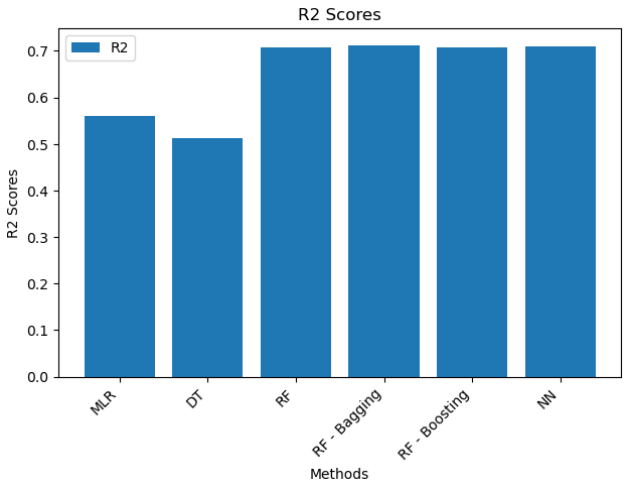
Training time:
Comparison across model configurations.

RMSE and R^2 scores:
Metrics for model accuracy.

method	time	RMSE	R2
MLR	0.002476	0.079951	0.560002
DT	0.011435	0.084213	0.511834
RF	0.617961	0.065176	0.707601
RF-Bagging	3.723051	0.06461	0.712651
RF-Boosting	17.543302	0.065192	0.707451
NN	0.576281	0.064962	0.709511







Results like this can inform which model has relatively high accuracy and low training cost.

OAK RIDGE National Laboratory

Summary and Conclusion

OAK RIDGE National Laboratory

Summary and Conclusion

Key Takeaways:

- High-quality synthesis of leaf-level data produce an “AI-ready” dataset of leaf-level fluxes
- User-friendly ML framework utilizes this “AI-ready” dataset
- Framework enables rapid experimentation of different data-driven ML recipes for predicting vegetation dynamics

Future Work:

- Build data-driven ML modules to plug into Earth System Models
- New “Hybrid” models will utilize AI-ready dataset and data-driven ML modules to supplement physical and empirical formulations
- This can make ESM simulations more accurate and efficient

Thanks



Contact:
massoudec@ornl.gov

OAK RIDGE National Laboratory

Synthesis of leaf-level photosynthesis data

Generating a synthesis dataset according to ESS-Dive leaf-gas-exchange guidelines

Minimum required variables based on leaf gas exchange data type:

variableName	survey	ACi curves	Photosynthetic parameters derived from ACi curves	Vcmax from one-point	AQ curves	Photosynthetic parameters derived from AQ curves	Dark adapted respiration
A	x	x		x	x		x
Ci	x	x		x	x	x	
CO2s	x				x	x	
gsw	x						
Patm	x	x	x	x	x	x	x
Qin	x	x	x	x	x		x
RHs	x	x	x	x	x	x	x
Tleaf	x	x	x	x	x	x	x
Fitted parameters			x	x		x	

OAK RIDGE National Laboratory

Synthesis Workflow

1. Initial review of source data: identify variables, units, errors, no-data values, constants, encoding inconsistencies, etc.
    a. Create graphs and data summaries for each variable
2. Clean up source header, description, and unit syntax
    a. Syntax standards derived from ESS-Dive leaf-gas-exchange guidelines
3. Create a lookup table for converting source header/description/unit to standard naming schema (.csv)
    a. EXAMPLE: leaf_temp, leaf temperature, F → Tleaf, Leaf surface temperature, C
4. Clean the data values
    a. Remove leading/trailing spaces, lowercase on all categorical data
    b. Set all identified no-data values to None
    c. Merge duplicate values (e.g., convert 'li-cor 6400' and 'licor-6400' to ‘licor 6400’
    d. Convert text in numeric columns into numeric (e.g., convert 0.5–2 meters to midpoint)
    e. Standardize year, month, day, time formatting to YYYY, MM, DD, HH:MM:SS, respectively
    f. Assign a plant identifier to each row based on dataFile, dataContributor, siteIdentifier, and species
    g. Create a lookup table for converting a species name to a standard species name (.csv)
        i. LATER: have an expert assign an acceptedName, nameLevel, nameAdjudicator, and nameAuthority
5. Curate source data contextual information (.csv)
    a. Citations, links, authors, etc.

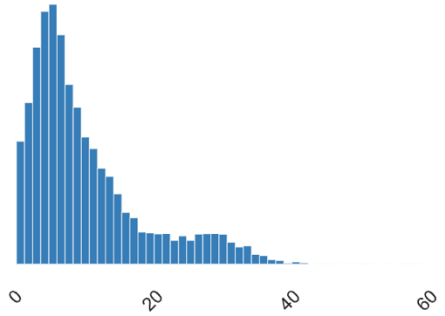
OAK RIDGE National Laboratory

A

Real number (ℝ)

HIGH CORRELATION

Distinct	7058	**Minimum**	0.0101
Distinct (%)	46.6%	**Maximum**	59.08
Missing	0	**Zeros**	0
Missing (%)	0.0%	**Zeros (%)**	0.0%
Infinite	0	**Negative**	0
Infinite (%)	0.0%	**Negative (%)**	0.0%
Mean	10.140983	**Memory size**	236.9 KiB



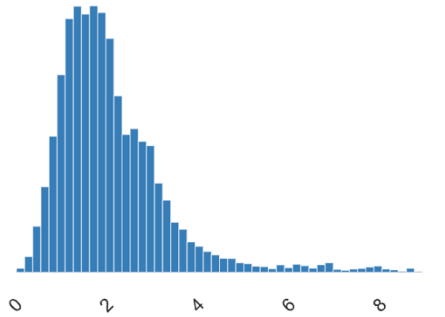
More details

VPDleaf

Real number (ℝ)

HIGH CORRELATION

Distinct	5861	**Minimum**	0.001
Distinct (%)	38.7%	**Maximum**	8.92
Missing	0	**Zeros**	0
Missing (%)	0.0%	**Zeros (%)**	0.0%
Infinite	0	**Negative**	0
Infinite (%)	0.0%	**Negative (%)**	0.0%
Mean	2.1057845	**Memory size**	236.9 KiB



More details