

Supplementary Material for

Space-based mapping of pre- and post-hurricane mangrove canopy heights using machine-learning with multi-sensor observations

Boya Zhang, Daniel Gann, Shimon Wdowinski, Chaohao Lin, Erin Hestir, Lukas Lamb-Wotton,
Khandker S. Ishtiaq, Kaleb Smith, Yuepeng Li

Content

Figure S1

Note S1

Figure S2-3

Table S1

Figure S4-6

Note S2

Figure S7

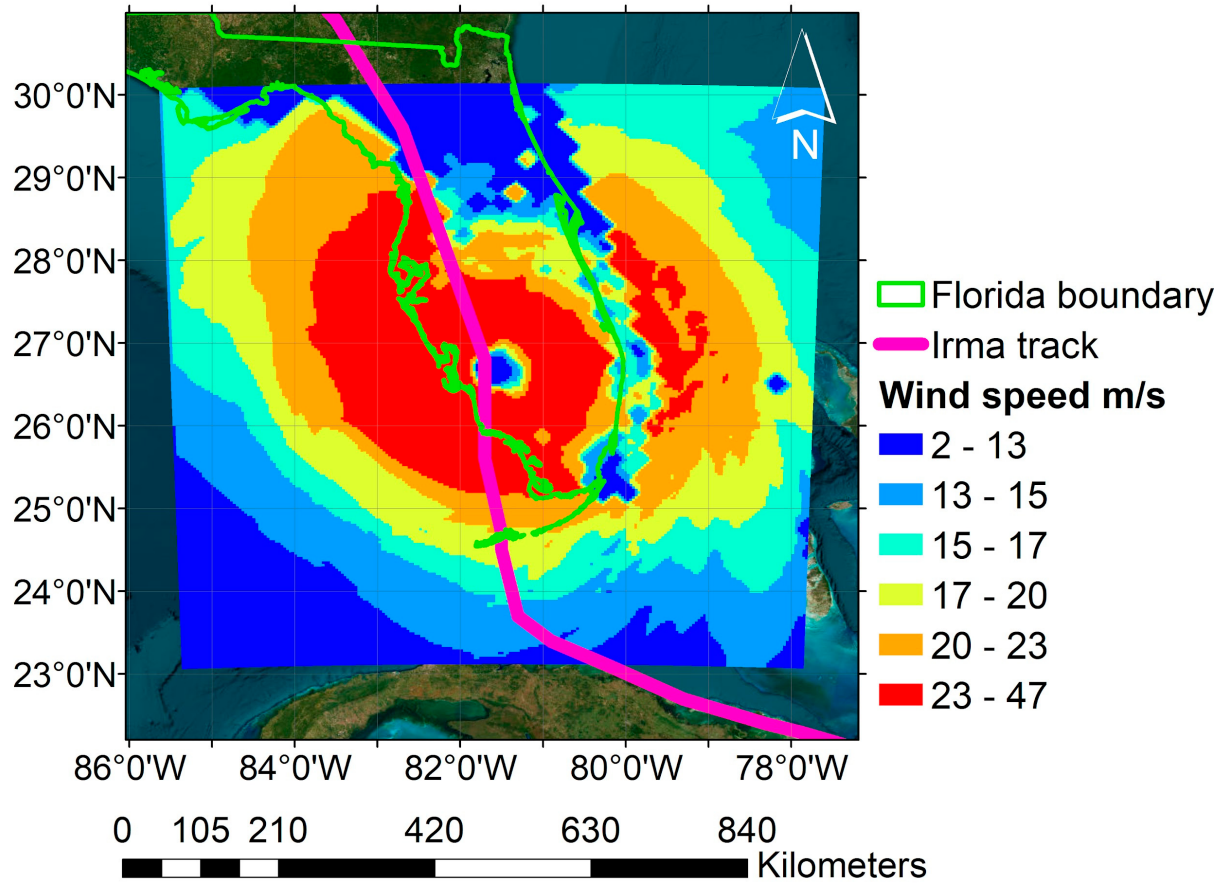


Figure S1. Hourly wind speed map for Hurricane Irma on September 10, 2017, between UTC 23:00 - 24:00. The magenta line shows the track of Hurricane Irma, and the green line represents the boundary of Florida state. The wind field was computed from Hurricane Weather Research and Forecast (HWRF) model, incorporating the Cyclone Global Navigation Satellite System (CYGNSS) data (Cui et al. 2019) [1].

Note S1. The machine learning and deep learning methods applied in this study

We briefly describe the mechanisms and optimization methods of the other six machine/deep learning models used in this study. The six models are Multiple Linear Regression, Decision Tree, Gradient Boosting, Support Vector Machine, Stacking Ensemble, and Multilayer Perceptron.

Multiple Linear Regression (MLR) is a statistical method that is used to model the relationship between a dependent variable and multiple independent variables by fitting a linear equation to observed data. Fitting a linear regression model involves finding the values of the coefficients that minimize the difference between the predicted and actual values of the dependent variable. This is typically done using a method called least squares, which minimizes the sum of the squares of the errors. In this work, we implement Multiple Linear Regression (MLR) using the Linear Regression class from the scikit-learn library.

Optimization: The basic form of this model does not include any hyperparameters that require tuning.

Decision trees offer a practical approach to data analysis and its structure includes: (1) root node as the first node of the tree, where the data is divided based on a certain feature, (2) internal nodes that test a condition or feature, leading to further branches based on the outcome of the test, (3) branches that represent a decision rule that leads to the next node based on the outcome, (4) leaf nodes or terminal nodes, representing the final output or the decision made after

computing all attributes. We used the DecisionTreeRegressor class from Scikit-learn to build our regression model.

Optimization: We applied grid search over the hyperparameters. Specifically, we varied max_depth (the maximum depth of the tree) from 1 to 20, min_samples_split (the minimum number of samples required to split an internal node) from 2 to 10, and min_samples_leaf (the minimum number of samples required to be in a leaf node) from 1 to 10. Through this process, we found that the best-performing model is achieved with a max_depth of 11, min_samples_split of 2, and min_samples_leaf of 1. We used squared error as the criterion for loss function.

Gradient Boosting is a method of creating a strong predictor from an ensemble of weak predictors. It involves training models sequentially, where each new model makes up for the deficiencies in the previous models. Gradient Boosting specifically addresses this by focusing on minimizing the loss, or error. Typical Gradient Boosting involves two main components, including base learner, where decision trees are commonly used as the base learner in Gradient Boosting. These trees are constructed in a greedy manner, selecting the best-split points based on purity scores like variance reduction for regression. The other main component is the additive model, which Gradient Boosting builds in a stage-wise fashion. It starts with a base model and sequentially adds new trees that correct the residuals or errors made by the previous trees.

Optimization: We fixed the min_samples_split at 2 and min_samples_leaf at 1. A grid search is then performed across the following hyperparameters: learning rate [0.1, 0.01, 0.001, 0.0001], number of trees [100,200,500,1000,2000], max tree depth (ranging from 2 to 11), subsample (ranging from 0.1 to 1.0), and regularization parameters (ranging from 0.1 to 1.0). Based on the

grid search results, we finalized the hyperparameters as follows: learning rate = 0.1, number of trees = 500, maximum tree depth = 3, subsample = 0.9, and regularization parameter = 1.0.

Support Vector Machine (SVM) regression, often referred to as Support Vector Regression (SVR), is a type of Support Vector Machine used for prediction tasks. SVR can use the kernel trick to handle nonlinear relationships. By mapping input features into high-dimensional space, SVR can find a linear hyperplane in this space, which corresponds to a nonlinear regression in the original feature space. SVR can be very effective in high-dimensional spaces while choosing and tuning the right kernel and regularization term can be challenging. In addition, epsilon is an essential parameter that controls the width of the tube within which predictions are not penalized.

Optimization: We used the radial basis function as the kernel for our model and performed a grid search to optimize the regularization parameter (ranging from 0.1 to 10.0) and epsilon (ranging from 0.1 to 1.0). After the search, we selected 1.0 for the regularization parameter and 0.1 for epsilon as our final hyperparameters.

Stacking combines multiple prediction models to generate a new model, often resulting in better predictive performance than any single constituent model. The main idea behind stacking is to learn how the initial models' predictions can be combined to make a final prediction more accurately. Stacking involves base models, which are trained with the full dataset. These models can be of different types (e.g., decision trees, support vector machines, neural networks). The next step is meta-model training based on the outputs of the base models. The actual output values from the training dataset are used as the target. This allows the meta-model to learn how to combine the base models' predictions best to make a final prediction. In summary, the base

models first predict new data, and then the meta-model uses these predictions as inputs to make the final prediction. This study utilized K-Neighbors, Decision Tree, and SVM as base models and set Linear Regression as meta-model.

Optimization: Based on the previous grid search, we fixed the parameters of the Decision Tree and SVM models to match those from earlier experiments. In this phase, we conducted a grid search specifically over the parameters of the k-nearest neighbors (k-NN) model. We explored varying the number of neighbors, ranging from 2 to 20. For this grid search, we used Euclidean distance (L2 norm) as the distance metric.

A Multilayer Perceptron (MLP) is a class of feedforward artificial neural networks that consists of at least three layers of nodes: an input layer, one or more hidden layers, and an output layer. MLPs can learn to model non-linear relationships between inputs and outputs, making them suitable for a wide range of complex tasks.

Optimization: We selected the optimal configuration after conducting a grid search over various hyperparameters—number of hidden layers [10, 20, 50, 100, 500], activation functions [ReLU, tanh, logistic], learning rate [0.1, 0.01, 0.001, 0.0001], and L2 regularization terms [0.1, 0.01, 0.001, 0.0001, 0.00001]. The final model was set with 100 hidden layers, ReLU activation function, a learning rate of 0.001, and an L2 regularization term of 0.0001.

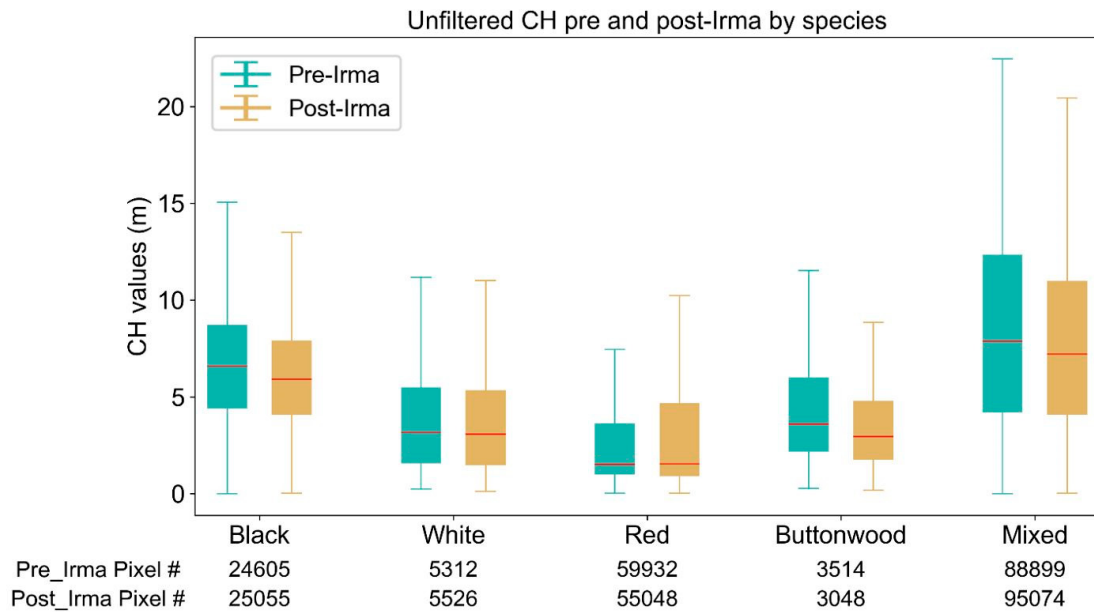


Figure S2. Boxplot for the LiDAR-derived pre- and post-Irma CHM by species from the *DS0* unfiltered dataset, including CHM values lower than 2 m. Red line indicates the median value; the boxes represent the interquartile range between the first quartile (25th percentile) and the third quartile (75th percentile); the whiskers extend from the edges of the box to the smallest and largest values within 1.5 times the interquartile range, respectively.



Figure S3. Comparison between the median, 75% and 25% percentile of SAR time series and optical feature values before and after Hurricane Irma. Darker color represents pre-Irma values and lighter color post-Irma values.

Table S1. Mean and standard deviation values of RMSE in meters for the two-time five-fold cross validation with seven machine/deep learning models. All models used the same training/testing datasets from the two-time, five-fold cross-validation.

ML/DL Method	Spatial Transfer Learning		Temporal Transfer Learning
	Pre-Irma	Post-Irma	
Random Forest	1.93±0.01	1.80±0.01	4.06±0.03
MLR	2.62±0.01	6.08±1.21	4.13±0.62
Decision Tree	2.05±0.02	1.86±0.02	4.05±0.23
Gradient Boosting	2.96±0.21	3.07±0.60	4.15±0.15
SVM	3.13±0.01	4.04±0.01	4.88±0.08
Stacking Ensemble	2.16±0.01	1.81±0.01	3.59±0.16
MLP	2.34±0.07	2.77±0.43	6.32±0.89

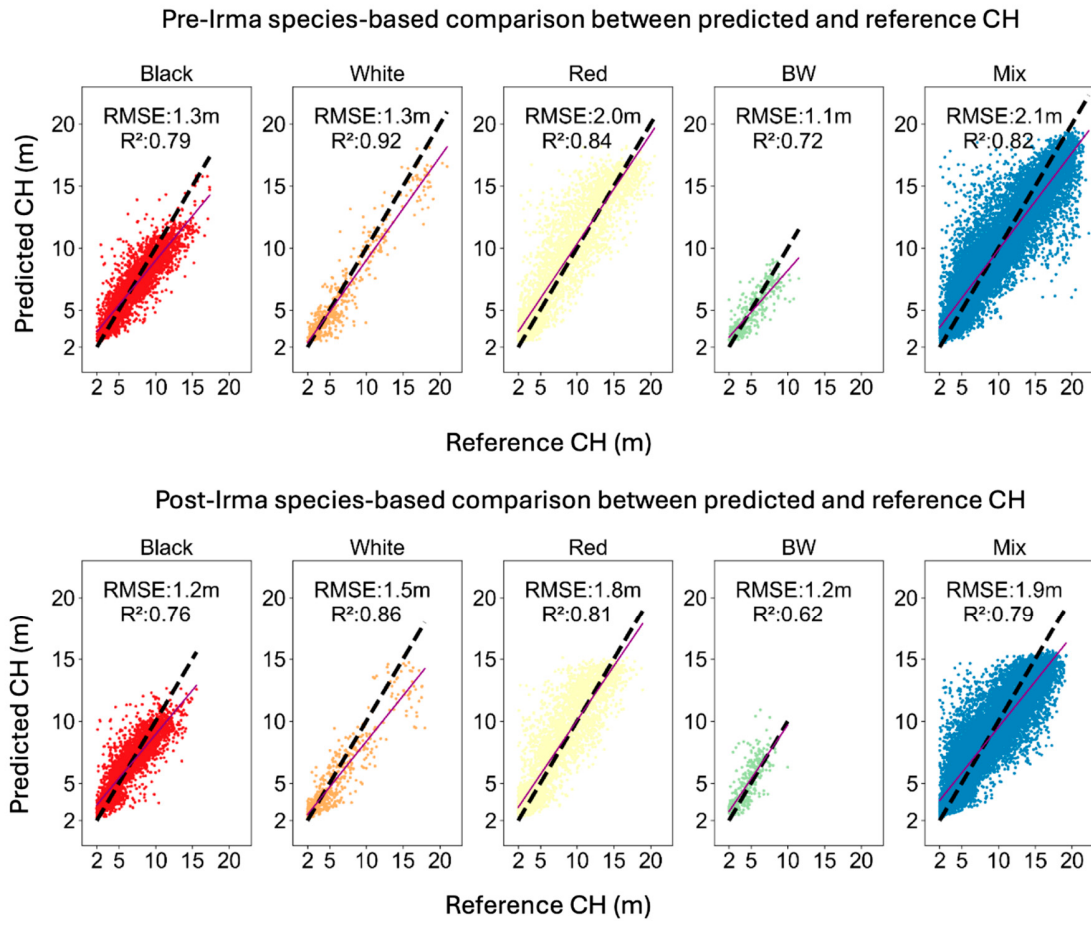


Figure S4. Species-specific plot between reference and predicted CH from spatial transfer learning. The combined plot of all species was displayed in Figure 5 in the manuscript.

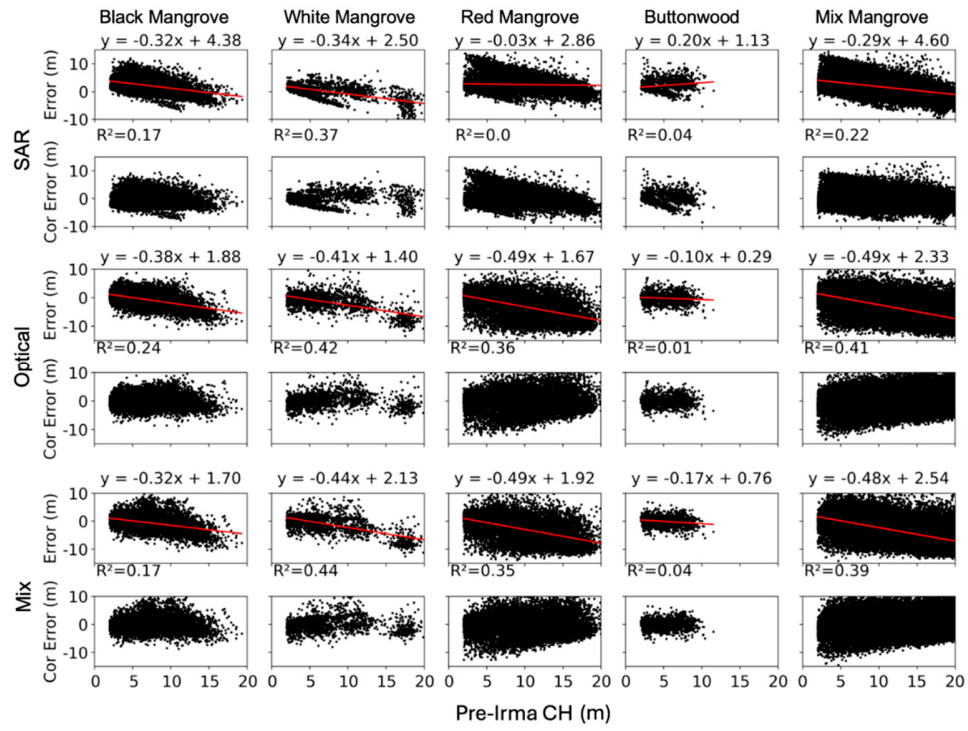


Figure S5. Scatterplots of temporal transfer learning errors and corrected errors versus pre-Irma CH in meter unit by species from a representative test dataset with SAR, optical, and mixed feature datasets, respectively.

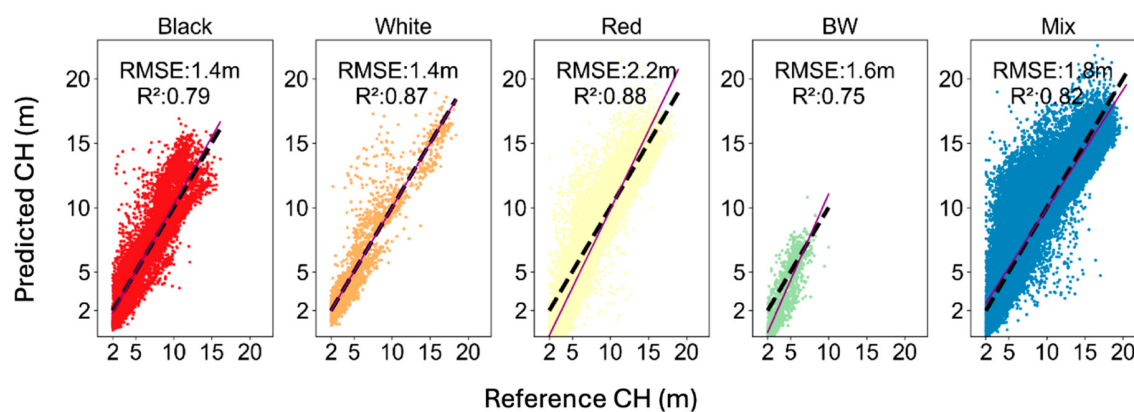


Figure S6. Species-specific plot of predicted and reference CH from temporal transfer learning.

The combined plot was displayed in Figure 8a.

Note S2 Evaluating SAR signals sensitivity at the ground level

To better comprehend if the SAR signals are able to reach the ground level (Fig. 7), we extracted water level gauge measurements and corresponding pixel backscatter time series across both pre- and post-Irma periods from a long-term ecological research site “Shark River Slough 6” (<https://fce-lter.fiu.edu/data/core/metadata/?packageid=knb-lter-fce.1168.13>) located in the mangrove forest approximately 150 m from the channel edge of Shark River. Considering significant sub-daily water level variations, we only used the hourly measurement for 18:00 pm local time (Eastern Standard Time) according to Sentinel-1A ascending path 48 that passed the Everglades around 6:30 pm. Backscatter was more sensitive to water level variations during the post-Irma period (Fig. S9c, d) because SAR signals can penetrate through open canopy to interact with water surfaces and tree trunks, leading to double-bounce scattering and positive linear relationships between backscatter and water levels. Backscatter values are indicative of properties within both the canopy (from direct scattering) and trunk layers (double-bounce scattering), hence yielding relatively higher variable importance for the post-Irma period. However, during the pre-Irma period, backscatter primarily originated from the top canopy layer (Fig. S2), which limits its ability to reflect the characteristics of the trunk layers and, consequently, with relatively lower variable importance during pre-Irma period.

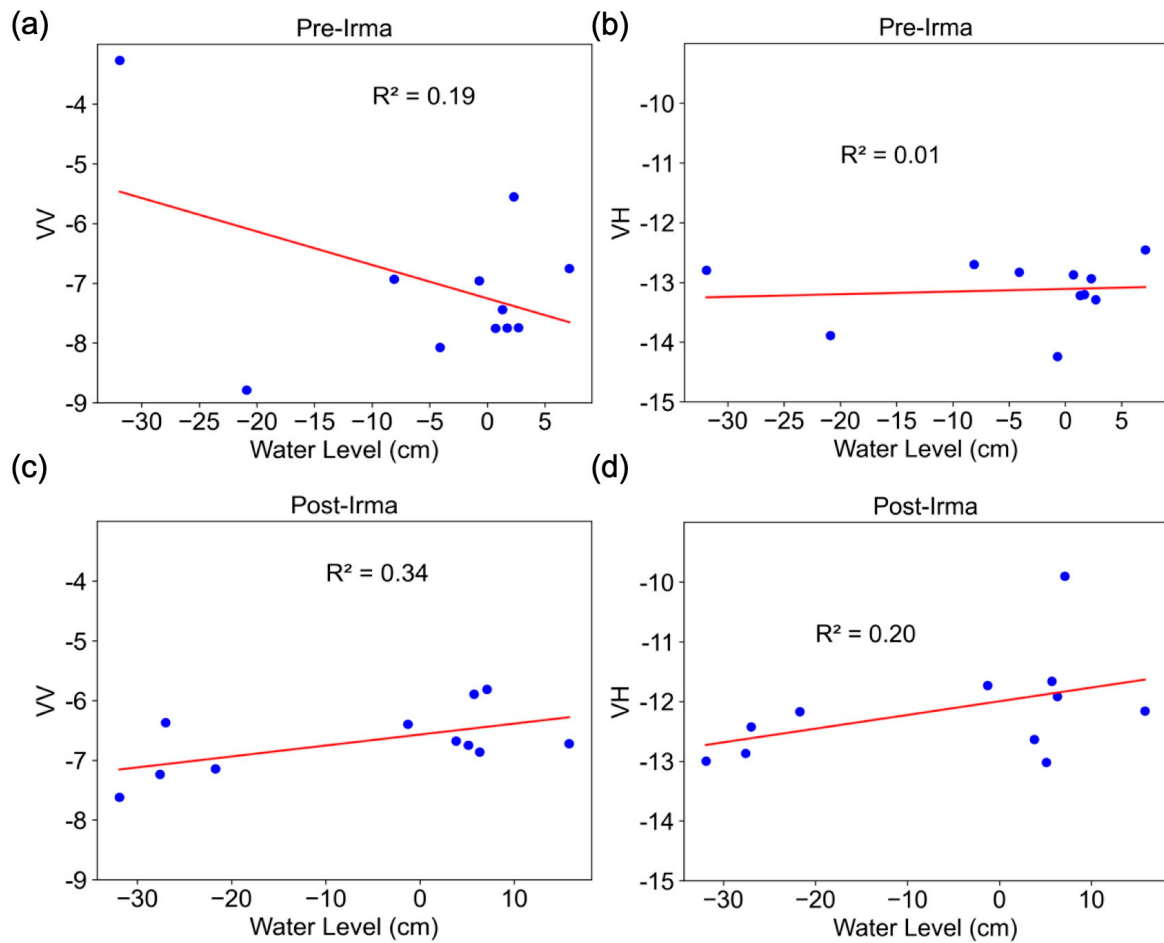


Figure S7. Scatterplot of SAR backscatter in VV and VH channels versus water depth relative to the ground surface for pre-Irma (a, b), and post-Irma (c, d). Water level data comes from a long-term ecological research site station (Shark River Slough 6) (<https://fce-lter.fiu.edu/data/core/metadata/?packageid=knb-lter-fce.1168.13>) located in the mangrove forest approximately 150 m from the channel edge of Shark River. Pre-Irma results (a, b) show weak correlation between backscatter and water level because backscatter is primarily from the top canopy layer, whereas, for post-Irma period, SAR signals reach the ground level and are sensitive to water level changes.

Reference

- [1] Cui, Z., Pu, Z., Tallapragada, V., Atlas, R., & Ruf, C. S. (2019). A preliminary impact study of CYGNSS ocean surface wind speeds on numerical simulations of hurricanes. *Geophysical Research Letters*, 46(5), 2984-2992.