Supporting Information for

Estimating spatiotemporally continuous snow water equivalent from intermittent satellite track observations using machine learning methods

Xiaoyu Ma[1], Dongyue Li[1,2], Yiwen Fang[2], Steven Margulis[2], Dennis P. Lettenmaier[1,2]

[1]Department of Geography, University of California, Los Angeles

[2]Department of Civil & Environmental Engineering, University of California, Los Angeles

Contents of this file

Introduction

This supporting information contains:

- Ancillary information on the study domain including physiographic characteristics across the watershed (Figure S1).

- The description of mathematical structure and working principles of statistical and machine learning methods used in this track-to-area (TTA) transformation. (Text S1, Figure S2, Figure S3).

- Results of feature sensitivity test based on statistical and ML methods in the extremely dry year (WY2015; Figure S4), the normal year (WY2008; Figure S5) and the extremely wet year (WY2017; Figure S6).

- Histogram of elevations of the pixels on the hypothetical ground tracks (Figure S7).

- Changes in MAE (m) of the MVLR-, RF-, SVM-, and DNN-based inferred April 1st SWE in dry, normal, and wet years with the number of ground tracks (Figure S8).
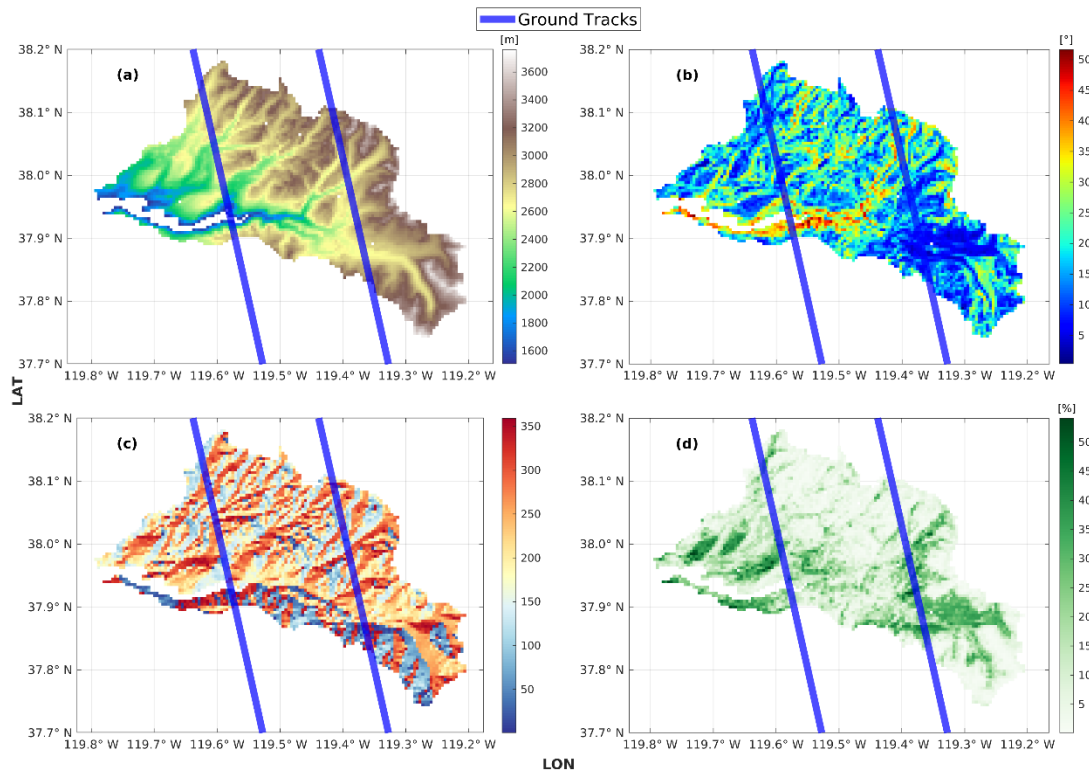
Figure S1. Elevation (a; in m), slope (b; in °), aspect (c), and fractional vegetation cover (d; in %) in the Upper Tuolumne River Basin. Elevations in the watershed range from around 700 m to approximately 3900 m. Slopes are distributed between 0° and larger than 50°. The terrain surface mostly has NW and SE facing aspects. Fractional vegetation cover ranges from 0% (in high-elevation areas) to up to 60% (in low-elevation areas).

Text S1.

Mathematical structure of multivariate linear regression

Multivariate linear regression (MVLR) is a statistical method based on assumed linear relationships between multiple independent variables (input variables) and one dependent variable (also termed here the target variable). In this project, the input variables are the meteorological forcings and static land cover features; the target variable is SWE.

It assigns a regression coefficient for each input variable and all coefficients are optimized together based on an algorithm that minimizes the sum of squares of the differences between observations and model predictions (least squares).

The MVLR structure takes the following form:

$$y = a + b_1 x_1 + b_2 x_2 + \cdots + b_n x_n \tag{1}$$

where y is the target (dependent) variable, a is a constant coefficient or intercept of the regression, b_i (i = 1, 2, 3, …, n) are regression coefficients, x is a vector of input (independent) variables, and ε is the model error.

Machine Learning Algorithms

1. Random Forest algorithms

In the random forest model, each tree is built using a deterministic algorithm by selecting a random set of variables and training samples. The basic building units of random forest are an ensemble of decision trees that split a subset of features on each split (Kuter, 2021). RF employs a series of decision trees to achieve sufficient accuracy for final prediction (Liu et al., 2020). The final prediction results are the weighted average of multiple selected decision trees. The selection of decision trees is carried out by voting. The higher the degree of repetition, the higher the contribution to the RF model.

The RF regression algorithm works as follows: (1) Extract N_{tree} bootstrap samples from the original training data set. (2). Grow a regression tree for each bootstrap sample. One-third (the default fraction for the number of different predictors tested at each node was 1/3) of the predictors are randomly sampled and the best split was selected from among those variables. (3). Predict new data by averaging the predictions from the Ntree regression trees. RF regression is appealing in scientific and engineering applications mainly because (1) RF is relatively robust to outliers (outliers in predictors are handled by binning them); (2) RF models can be trained efficiently since the algorithm works on only a subset of features in training inputs; (3) RF models are self-explainable because we can measure the relative importance of each predictor using the "out-of-bag" data (Cutler et al., 2012).

In this project, the "out-of-bag" prediction ("OOBPrediction" and "OOBPredictorImportance" in the "TreeBagger" function in MATLAB) was on during the

modeling tuning process and was off after we finished the construction of the RF model.
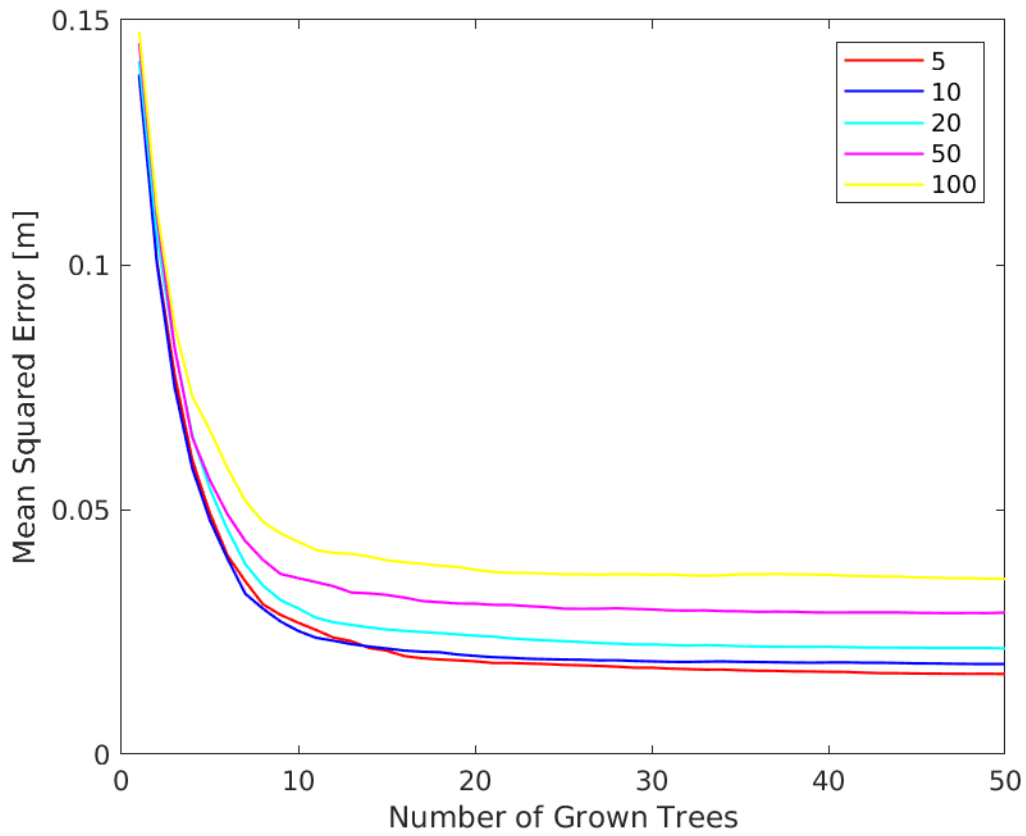
Figure S2. The change of "out-of-bagger" mean squared errors with the number of growing trees as the minimum number of observations per tree leaf is 5, 10, 20, 50, 100, respetively.
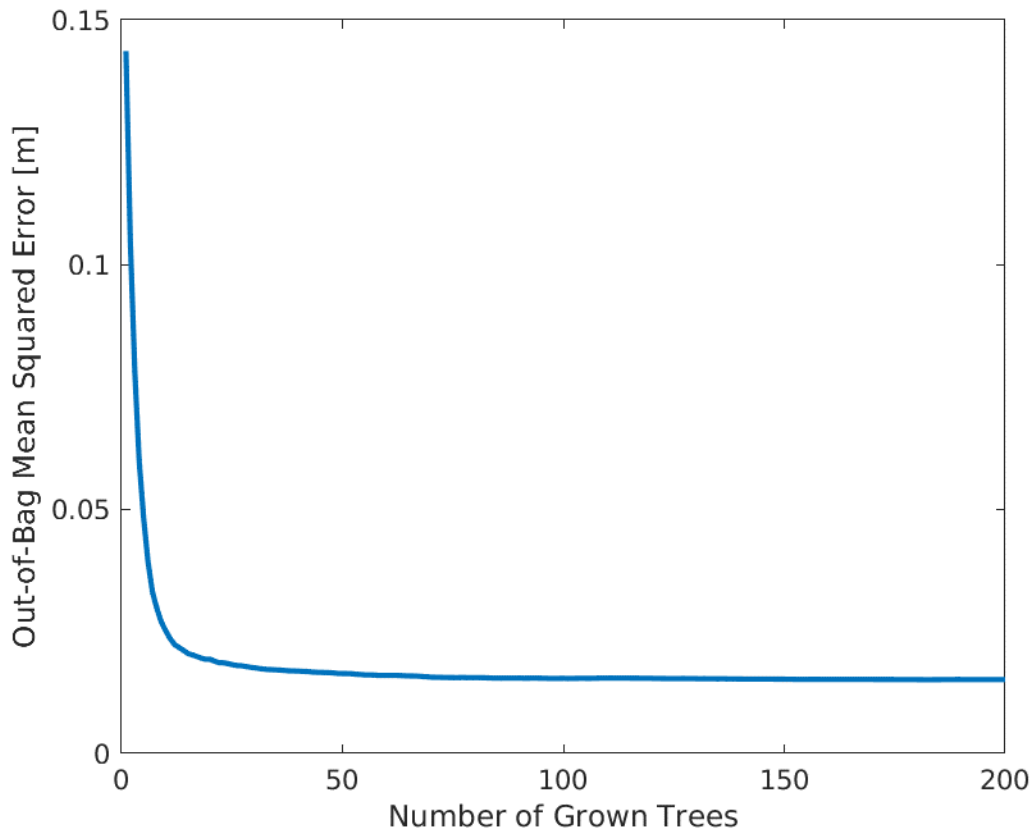
Figure S3. The change of "out-of-bag" mean squared errors with the number of growing trees as the minimum number of observations per tree leaf is 5.

2. Support Vector Machine

Originally, support vector machines (SVM) was designed as a linear binary classifier that aims to find the optimal decision boundary that (1) maximizes the margins between classes and (2) minimizes the misclassification errors (Vapnik, 2000). For regression-based SVM, the basic logic behind the learning task is to find a function f(x) that has the universal minimum deviation from the measured response values for the full range of observations (Vapnic, 1998). Using MATLAB "fitrsvm" function, we carried out 10-fold cross validation to optimize two hyperparameters: (1) the kernel function, which specifies the

method used to transform inputs to the required target, and (2) the kernel scale: the software divides all elements of the predictor matrix X by the value of kernel scale. Then, the software applies the appropriate kernel norm to compute the Gram matrix. For (1), we chose "Gaussian" from the linear, polynomial, and Gaussian function. For (2), we selected "auto" from 1 and "auto".

SVM regression is attractive because: (1) it does not require assumptions about the underlying probability distribution of the data; and (2) it can handle large and complex training data since the position of the decision surface is determined by only a portion of the training dataset and has no relation with the training data set's dimension (Mathur and Foody, 2008; Mountrakis et al., 2011).

3. Deep neural network

Here we constructed a seven-layer neural network with 10, 9, 8, 7, 6, 5, and 4 neurons in each hidden layer, respectively based on the MATLAB deep learning toolbox. Here, we chose Rectified Linear Units (ReLu) as the activation function in each hidden layer. In the process of model training, the training data are given to each layer, and the neuron weights are adjusted to achieve the minimum error between the model output and the observations used for training. A portion of training data (not present in the training process) remain for model validation. Also, different inputs (not present in the training and validation phases) are used for model testing by tuning model parameters (primarily weights). Here, the percentage of the training, validation, and test data set is 75%, 15%, and 15%, respectively.

The output of a neuron can be expressed as follows:

$$y_i = \varphi\left(\sum W_{ji}\, x_i + b_i\right) \qquad (2)$$

Where φ is the activation function (herein, ReLU function) for the jth neuron in the hidden layer, y_i is the output, W_{ji} is the weight of the jth neuron, x_i is the input, and b_i represents the bias vector.

The cost function used in this network is the mean squared error (MSE) and the Levenberg-Marquardt algorithm is adopted to solve the problem of overfitting.
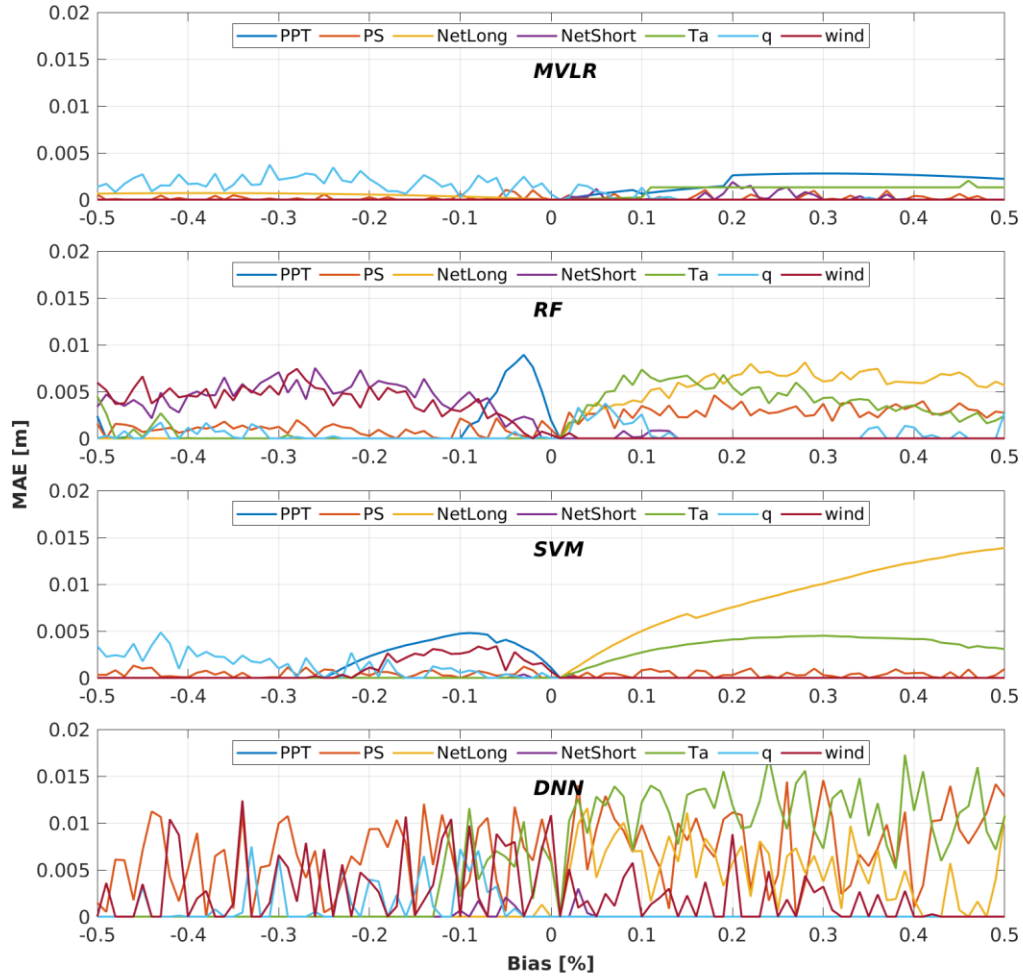
Figure S4. Changes of MAE (m; relative to no-bias situation) of the inferred April 1st SWE in WY2015 (extremely dry year) with biases in the forcings of the training data set. The modeled SWE results are from MVLR (first row), RF (second row), SVM (third row), and DNN (fourth row).
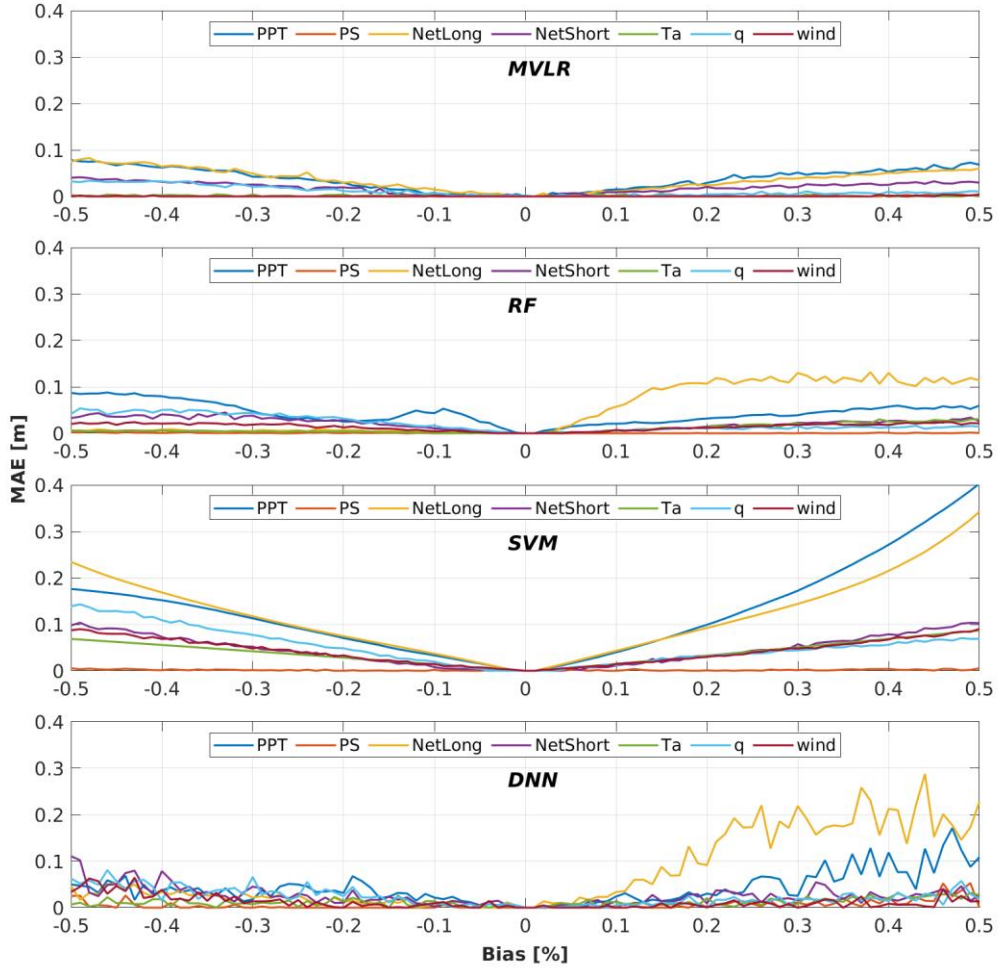
Figure S5. Changes of MAE (m; relative to no-bias situation) of the inferred April 1st SWE in WY2008 (typical normal year) with biases in the forcings of the training data set. The modeled SWE results are from MVLR (first row), RF (second row), SVM (third row), and DNN (fourth row).
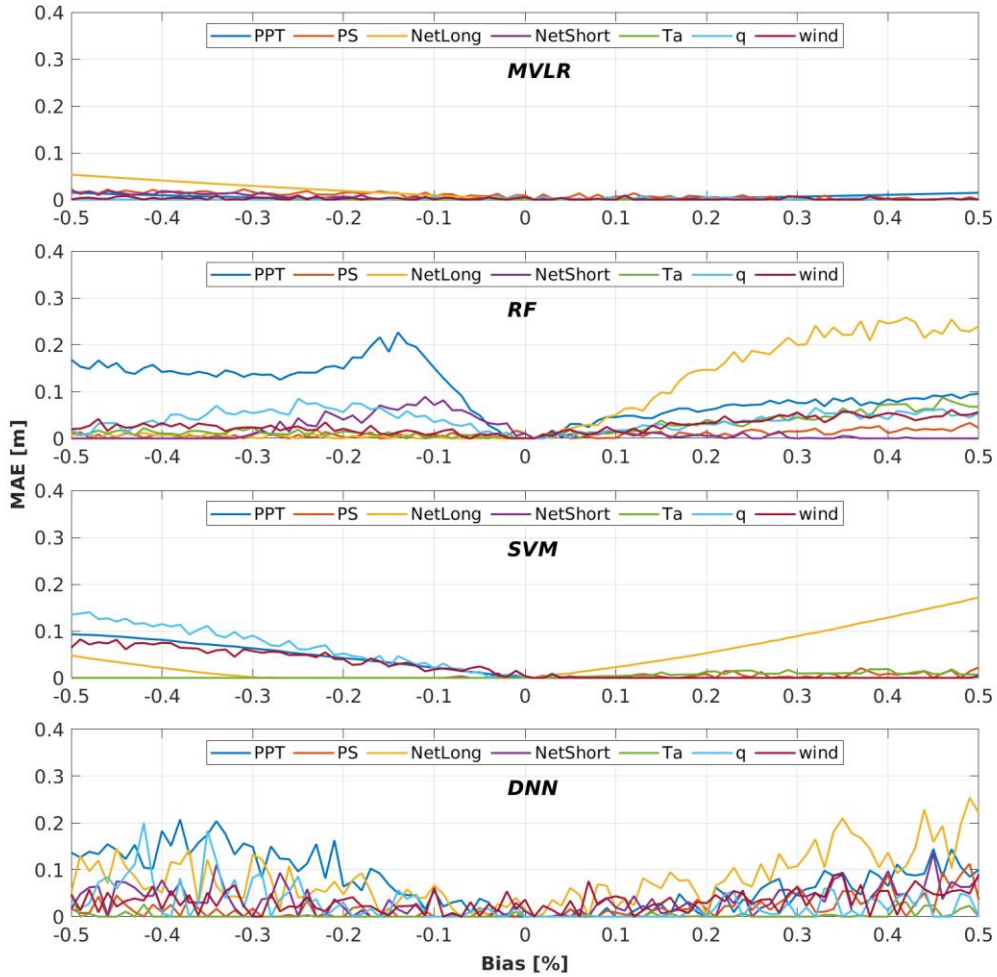
Figure S6. Changes of MAE (m; relative to no-bias situation) of the inferred April 1st SWE in WY2017 (extremely wet year) with biases in the forcings of the training data set. The modeled SWE results are from MVLR (first row), RF (second row), SVM (third row), and DNN (fourth row).

MVLR has a relatively flat trend in MAE as the absolute biases become larger. As the biases of each meteorological field increase from 0 to +/-50%, MAE from all sensitivity tests increase except for Ta and wind, which has a much smaller influence on the results of SWE estimates based on MVLR. PPT and NetLong are the two variables that have the

largest impacts on SWE estimation results. Positive biases of PPT and negative biases of NetLong and PPT can lead to larger errors than any positive/negative biases in other forcing fields. Generally, the values of MAE are less than 0.005 in the dry year and less than 0.1 m in the normal and wet years if the biases in meteorological inputs are up to +/- 50%.

RF is the algorithm that is least sensitive to forcing errors in the training processes among the three ML methods. Basically, the MAE is less than 0.15 m in WY2015 and WY2008 and less than 0.25 m in WY2017 even though larger biases (i.e., ±50%) are added. Errors in precipitation are the main source of SWE estimate errors. It is noticeable that positive biases in NetLong can lead to greater errors in the results. Thus, in the real application, quality control of NetLong is of key significance when employing RF for TTA SWE observation transformation. SVM is the ML algorithm that is most sensitive to the biases in the training data set. Except for wind speed, biases (negative or positive) in all the forcing fields cause relatively larger errors in SWE estimation. Especially, SVM-based TTA transformation is more sensitive to the biases in PPT and NetLong.
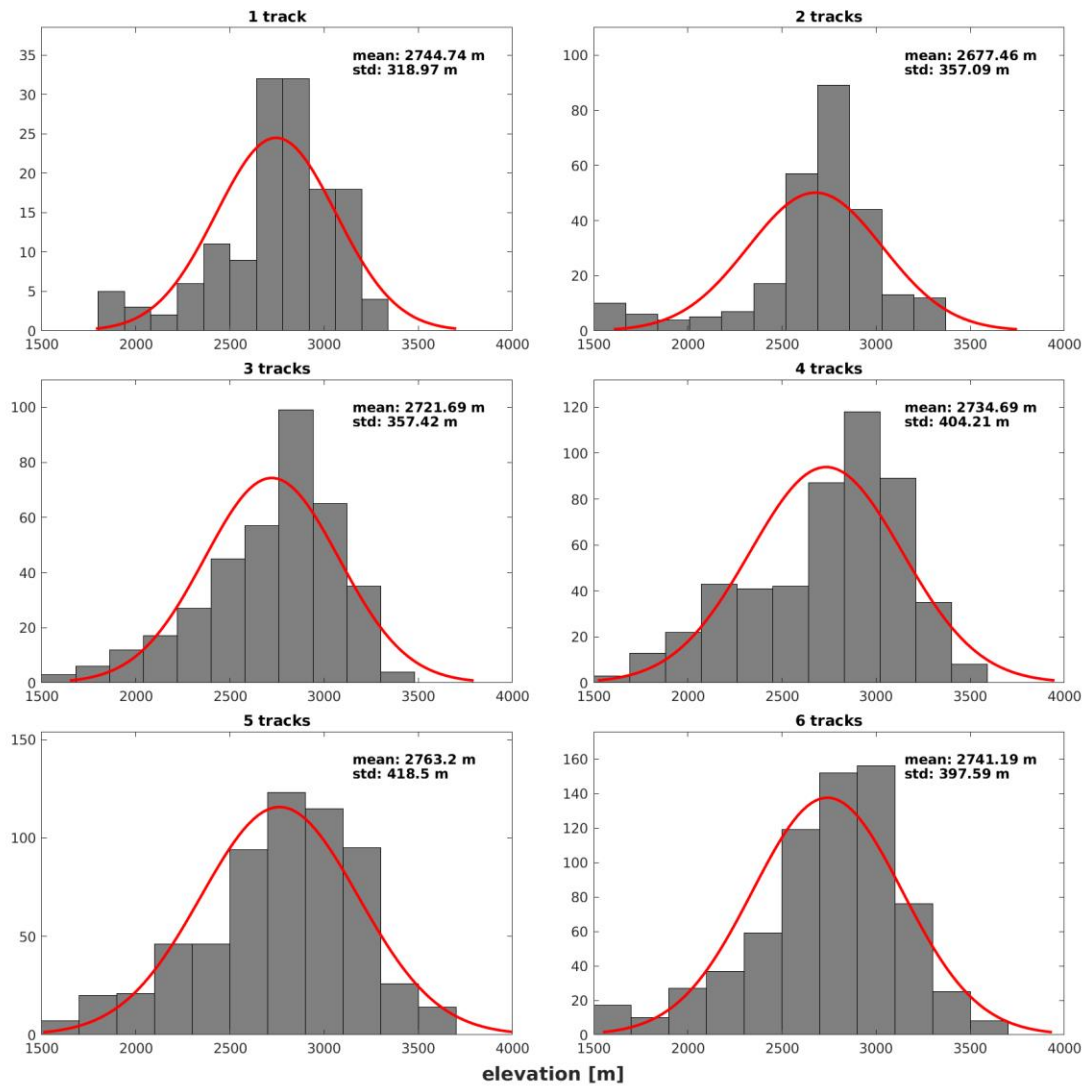
Figure S7. Histogram of elevations of the pixels on the hypothetical ground tracks. The red lines show the fitted normal distribution of elevations on ground tracks with mean and standard deviation (std) listed on the figures.
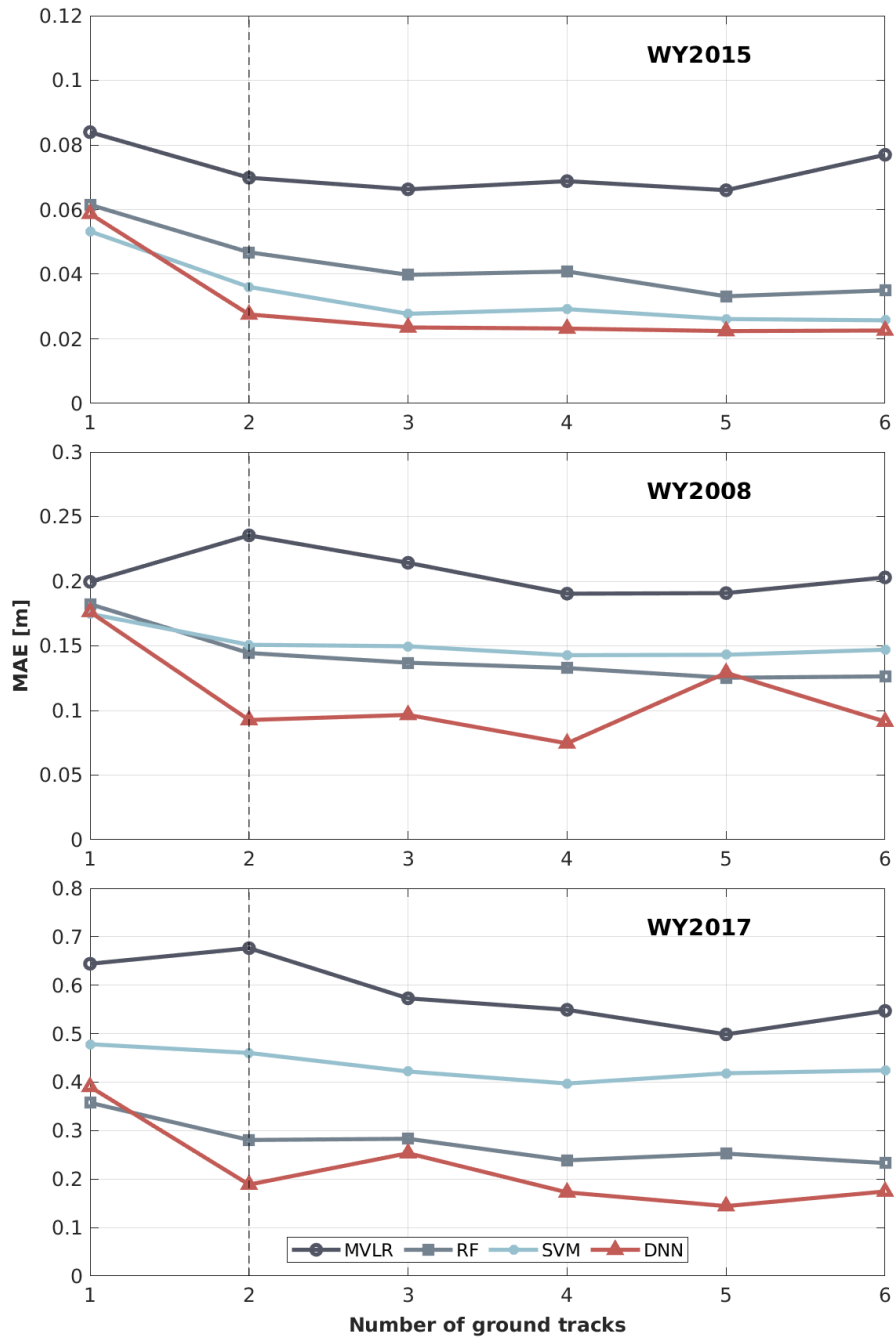
Figure S8. Changes in MAE (m) of the MVLR-, RF-, SVM-, and DNN-based inferred April 1st SWE in dry, normal, and wet years with the number of ground tracks

References

Cutler, A., Cutler, D. R., and Stevens, J. R.: Random Forests, in: Ensemble Machine Learning: Methods and Applications, edited by: Zhang, C. and Ma, Y., Springer US, Boston, MA, 157–175, https://doi.org/10.1007/978-1-4419-9326-7_5, 2012.

Kuter, S.: Completing the machine learning saga in fractional snow cover estimation from MODIS Terra reflectance data: Random forests versus support vector regression, Remote. Sens. Environ., 255, 112294, https://doi.org/10.1016/j.rse.2021.112294, 2021.

Liu, C., Huang, X., Li, X., and Liang, T.: MODIS Fractional Snow Cover Mapping Using Machine Learning Technology in a Mountainous Area, Remote. Sens., 12, 962, https://doi.org/10.3390/rs12060962, 2020.

Mathur, A. and Foody, G. M.: Crop classification by support vector machine with intelligently selected training data for an operational application, 29, 2227–2240, Int. J. Remote Sens., https://doi.org/10.1080/01431160701395203, 2008.

Mountrakis, G., Im, J., and Ogole, C.: Support vector machines in remote sensing: A review, ISPRS J. Photogramm., 66, 247–259, https://doi.org/10.1016/j.isprsjprs.2010.11.001, 2011.

Vapnik, V. N.: The Nature of Statistical Learning Theory, Springer New York, New York, NY, https://doi.org/10.1007/978-1-4757-3264-1, 2000.