Evaluation often machine learning methods for estimating terrestrial evapotranspiration from remote sensing

Corinne Carter and Shunlin Liang*

Department of Geographical Sciences, University of Maryland, College Park, MD USA
*corresponding author, sliang@umd.edu

Abstract: Remote sensing retrieval of evapotranspiration (ET), or surface latent heat exchange (LE), is of great utility for many applications. Machine learning (ML) methods have been extensively used in many disciplines, but so far little work has been performed systematically comparing ML methods for ET retrieval. This paper provides an evaluation of ten ML methodsfor estimating daily ET based on daily Global LAnd Surface Satellite (GLASS) radiation data and high-level Moderate-Resolution Imaging Spectroradiometer (MODIS) data products and ground measured ET data from 184 flux tower sites.Measurements of accuracy (RMSE, R^2, and bias) and run time were madefor each of ten ML methods with a smaller training data set (n = 7910 data points) and a larger training data set (n= 69,752 data points). Inclusion of more input variables improved algorithm performance but had little effect on run time. The best results were obtained with the larger training data set using the bootstrap aggregation (bagging) regression tree (validation RMSE = 19.91 W/m^2) and three hidden layer neural network (validation RMSE = 20.94 W/m^2), although the less computationally demanding random kernel (RKS) algorithm also produced good results (validation RMSE = 22.22 W/m^2). Comparison of results from sites with different ecosystem types showed the best results for evergreen, shrub, and grassland sites,

and the weakest results for wetland sites. Generally, performance was not improved by training with data from only the same ecosystem type.

Introduction

Evapotranspiration (ET), often expressed as an energy flux, the latent heat of evaporation (LE), is an important linkage between the surface energy and water balances and an indicator of vegetation health. Compared to the radiative elements of the surface energy balance, there is more uncertainty in LE measurements. Ground-based measurements are made at small scales with weighing lysimeters, and at scales of tens of meters to kilometers with flux towers andscintillometers. However, these measurements are sparse outside the northern hemisphere midlatitudes. Remote sensing data, reanalyses, and ground-based observations have been combined in a variety of ways to retrieve LE. Reviews of methods for obtaining LE through remote sensing are available inZhang et al. (2016), Wang and Dickinson (2012), and Kalma et al. (2008). Some of these methods (e. g. Wang and Liang 2008; Yao et al. 2011, 2013, 2015; Yebra et al. 2013; Helman et al. 2015) use statistical regression techniques.Carter and Liang (2018)evaluated a number of statistical regression formulas for obtaining LE.

Machine learning (ML) methods are means of extracting patterns from data sets with little prior knowledge of those patterns. The best-known ML methods include neural networks (NN), tree methods, and support vector machines (SVMs). The model tree ensemble technique has been applied to the problem ofdetermining global trends in LE by Jung et al. (2010). Multiple studies have been conducted using machine learning techniques for downscaling LE (Ke et al. 2017, 2016; Kaheil et al. 2008) and drought detection and forecasting (Rhee and Im 2017; Park et al. 2016). There are also a number of studies comparing the performance of different ML techniques

for obtaining LE. In these studies (eg.Deo et al. 2016; Dou and Yang 2018)no single ML method produced the best results. Most of these studies, with the notable exception of Jung et al. (2010),involve training to measurements of LE from a relatively small number of locations (20 or fewer). In previous studies where ML method comparisons are performed four or fewer methods are compared.

The goal of this study is to evaluate the utility of a range of ML algorithms for obtaining LE from remote sensing data on a global basis, and to evaluate their performance for different ecosystem types.

Data

The remote sensing data used in this study are Global Land Surface Satellite (GLASS) radiation data and Moderate-Resolution Imaging Spectroradiometer (MODIS) high-level data products. Ground-based Fluxnet tower site data were also used. The data variables, sources, and spatial and time resolutions for each data set used are listed in Table 1.

Table 1: Input and validation data used in this study

Abbreviation	Variable	Source	Frequency	Spatial resolution
LE	Surface latent heat	Fluxnet	Half-hourly, averaged to daily	Flux tower footprint
R_n	Net radiation at surface	Fluxnet	Half-hourly, averaged to daily	Flux tower footprint
DSR	Downward surface radiation	GLASS	Daily	5 km

PAR	Photosynthetically active radiation	GLASS	Daily	5 km
NDVI	Normalized difference vegetation index	MODIS	16-day, interpolated to daily	250 meters
EVI	Enhanced vegetation index	MODIS	16-day, interpolated to daily	250 meters
LAI	Leaf area index	MODIS	8-day, interpolated to daily	500 meters
FPAR	Fraction of photosynthetically adjusted radiation	MODIS	8-day, interpolated to daily	500 meters
Albedo	Albedo	MODIS	8-day, interpolated to daily	500 meters
NBAR	Nadir BRDF-adjusted reflectance	MODIS	8-day, interpolated to daily	500 meters

The GLASS data set (Liang et al. 2013, 2014)consists of radiative and biophysical parameters generated using data from multiple satellite sensors. The products used here are the downward shortwave radiation (DSR) and photosynthetically active radiation (PAR).

Several parameters obtained from MODIS were also used in this analysis: Normalized-difference vegetation index (NDVI) and enhanced vegetation index (EVI) (Didan 2015), leaf area index (LAI) and fraction of photosynthetically active radiation absorbed (FPAR) (Myneni and Knyazikhin 2015), surface albedo, and nadir BRDF-adjusted reflectance (NBAR) (Schaaf and Wang 2015a, 2015b). Subsets of all MODIS products used were generated centered on the coordinates of each flux tower site. All MODIS products were linearly interpolated to daily frequency.

Flux tower data were used for validation of the ML algorithms, and also for testing the effects of using remote sensing vs. ground-based radiation data as input. A total of 184 flux tower sites

were used, 119 from the Ameriflux network (http://ameriflux.lbl.gov) and 65 from the Fluxnet2015 data set (http://fluxnet.fluxdata.org/data/fluxnet2015-dataset/). The half-hourly LE and net radiation (R_n) variables from these data were pre-processed by removing all data days for which there were not at least 40 of 48 possible observations present, then averaging the remaining observations. A map of the site locations and information about their distribution across ecosystem types is given in Carter and Liang (2018).

A total of 79098 site-days of data were used, randomly partitioned twice into 7,910/ 35,594/ 35,594 and 63,278/ 7,910/ 7,910 site-days of training, validation, and test data, respectively.These training, validation, and test data sets were used in every case except where only data from individual ecosystem types was used. Each site-day includes a flux tower LE value associated with the remote sensing parameters retrieved at the site for that day. For purposes of this study, we treated the flux tower LE values as “ground truth”, although flux tower footprints vary and may not always coincide closely with the pixel size of the remote sensing data and also require adjustment to compensate for lack of energy balance closure.

Methods

In order to use the ML algorithms properly, it is necessary to adjust one or more tunable parameters for each algorithm. This is done by training the algorithms with a training data set for different parameter values and checking against a validation data set until the optimum parameter values are found, for example by minimizing RMSE for the validation data set. Once the optimization is performed, the optimized algorithm is checked against a test data set separate from the training and validation data sets. Timing of a single iteration of training with each

training data set and checking with the validation data set was performed for each algorithm as a feasibility check, since it is necessary to repeat this process tens to hundreds of times to tune the algorithms. Timing was conducted on a server with 24 6-core 3.33GHz Intel Xenon X5680 CPUs.

Fourteen ML algorithms were subjected to initial timing tests with the smaller training data set. Based on the results of this timing, 10 of the original algorithms were tuned with the smaller training data set. Of those 10, 8 were found to run with enough efficiency for systematic tuning with the larger training data set to be feasible. The 14 ML algorithms considered are listed in Table 2, with references to descriptions of each of the algorithms.

Table 2: Algorithms tested in this study.

Family	Full name	Abbreviation	Tuned with small training data set	Tuned with large training data set
Linear	Regularized linear regression*	RLR	Yes	Yes
	Least absolute shrinkage and selection operator regression*	LASSO	Yes	Yes
	Elastic net regularization*	ELASTIC	Yes	Yes
Kernel	Gaussian process regression*(Murphy 2012)	GPR	No	No
	Kernel ridge regression (Murphy 2012)	KRR	Yes	No
	Random kernel (Rahimi and Recht 2009, Perez-Suay et al. 2017)	RKS	Yes	Yes
	Variational	VHGPR	No	No

	heteroscedastic Gaussian process regression(Lazaro-Gredilla et al. 2014; Lazaro-Gredilla and Titsias 2011)			
Tree	Regression tree*	TREE	Yes	Yes
	Bootstrap aggregation (bagging) tree*	BAGTREE	Yes	Yes
	Boosted regression tree*	BOOST	Yes	Yes
Neural network	Standard neural network (1, 2, and 3 hidden layers)*	NN	Yes	Yes
	Extreme learning machine (Huang et al. 2006)	ELM	No	No
Support vector	Support vector regression* (Smola and Scholkopf 2004)	SVR	Yes	No
	Relevance vector machine(Thayananthan et al. 2006)	RVM	No	No

Algorithms marked with an asterisk (*) are described in Hastie et al. (2008).

Optimum values of the parameters are found by minimizing the root mean square error (RMSE) of the algorithm when applied to the validation data set. The coefficient of determination R^2 and bias were also used to characterize the correspondence of the modeled LE from different surface types. The implementation in Matlab of all of these algorithms, with the exception of the random kernel (RKS), was obtained from package “simpleR” (Lazaro-Gredilla et al. 2014). The RKS algorithm code was obtained from http://isp.uv.es/code/rks2017.html(Pérez-Suay et al. 2017).

Initially, one training/ validation iteration was timedfor each ML algorithm using a smaller training data set. Algorithms that took more than ten minutes for one iteration were removed from further consideration. The remaining algorithms were timed for one training/ validation

iteration with a larger training data set. Two of the algorithms (KRR and SVR) that were tractable with the smaller training data set became too computationally demanding with the larger training data set. These timing results aregiven in Results section 1.The linear regression, boost tree, and RKS were used to test the effects on accuracy and computation time of using different combinations of input variables, as discussed in Results section 2.

Once the most viable combinations of ML algorithms and input variables had been identified, each algorithm was tuned by varying all parameters of each algorithm independently. The optimal tuning parameters with respect to the validation data set (lowest validation RMSE) were applied to the test data set. Variation in algorithm performance with tuning of parameters and optimum algorithm performancesare shown in Results section 3.

The RKS, BAGTREE, and 2 and 3 hidden layer NNs were applied to each of seven ecosystem types' test data sets, first with the algorithms optimized using data from all of the sites, and then optimized using data from sites of the same ecosystem type only. These results are given in Results section 4.

Results

1. Initial time trials of ML algorithms

The time in seconds for each algorithm to run a single iteration of training and validation with allinput variables is shown in Table 3. If an algorithm took longer than 10 minutes to run a single iteration, it is labeled "prohibitive" and no further testing was done for that algorithm.

Table 3: Time in seconds for one iteration of training and validation for each algorithm.

Algorithm	Small training data set (7,910 data points)	Large training data set (69,752 data points)
RLR	0.0014	0.0087
LASSO	19.3208	118.1654
ELASTIC	21.8523	109.0231
GPR	prohibitive	
KRR	219.6609	prohibitive
RKS, D = 100	0.0945	1.1625
D = 400	0.3950	4.7896
D = 1000	1.1794	13.026
D = 4000	9.1198	91.2732
TREE	20.2009	351.7759
BAGTREE	15.9202	114.309
BOOST (200 trees)	3.1619	9.1393
NN, 1 HL, 5 neurons	4.0422	102.3333
1 HL, 30 neurons	6.2271	207.9746
2 HL, 5 x 5 neurons	5.5793	108.4992
2 HL, 10 x 10 neurons	6.0898	131.7211
2 HL, 30 x 30 neurons	10.0308	436.8679
3 HL, 5 x 5 x 2 neurons	4.6199	128.9298
3 HL, 10 x 10 x 10 neurons	7.8482	153.7699
3 HL, 50 x 5 x 2 neurons	12.849	prohibitive
3 HL, 150 x 30 x 10 neurons	249.671	prohibitive
ELM	prohibitive	
SVR	41.6029	prohibitive
RVM	prohibitive	
VHGPR	prohibitive	

Note: For RKS, “D” represents the number of random functions used. The number of hidden layers (HL) and neurons in each of the NN trials is also indicated.

2. Combinations of input variables

In order to test the effects on speed and accuracy of using different combinations of input variables, trials of a single training and testing cycle were done with the linear regression, boost tree, and RKS methods using the small training data set. Boost tree RMSE was found after optimizing the number of trees, but timing trials were done for 100 and 1000 trees. Results are summarized in Table 4. Generally, including more input variables produced similar or more accurate results at little additional computational cost. Using radiation information from surface measurements produced results of similar accuracy to using the GLASS radiation variables.

Table 4: Accuracy and timing tests for different combinations of input variables using the smaller training data set.

Variables	Linear regression RMSE (W/m^2)	Linear regression timing (s)	BOOST RMSE (W/m^2)	BOOST timing (100 trees) (s)	BOOST timing (1000 trees) (s)	RKS RMSE (W/m^2)	RKS timing (s)
R_n + NDVI	34.68	6.74×10^{-4}	32.52	2.71	28.25	31.71	0.17
R_n + NBAR	31.80	0.0032	29.18	3.15	31.11	28.10	0.20
R_n + NDVI + EVI + LAI + FPAR + NBAR + Albedo	31.78	0.0098	28.03	4.33	38.08	28.10	0.20

DSR + NDVI	33.83	8.69×10^{-4}	33.07	2.70	28.51	31.73	0.17
PAR + NDVI	33.67	6.72×10^{-4}	32.89	2.77	27.74	31.84	0.18
DSR + PAR + NDVI	33.66	0.0016	32.84	2.69	28.88	31.19	0.18
DSR + PAR + EVI	33.29	0.0013	32.23	2.94	28.96	30.41	0.18
DSR + PAR + NDVI + EVI	33.21	0.0014	31.62	3.04	28.27	29.78	0.18
DSR + PAR + FPAR	33.62	0.0014	32.75	2.99	27.83	31.09	0.17
DSR + PAR + LAI	34.47	0.0016	32.46	2.94	28.31	31.10	0.20
DSR + PAR + LAI + FPAR	33.63	0.0014	31.84	2.80	27.85	30.36	0.20
DSR + PAR + NDVI + EVI + LAI + FPAR	32.90	0.0030	30.53	3.28	31.55	29.14	0.20
DSR + PAR + Albedo	36.96	0.0015	35.85	2.70	26.92	35.49	0.17
DSR + PAR + NDVI + EVI + LAI + FPAR + Albedo	32.89	0.0028	30.26	3.26	31.31	28.95	0.18
DSR + PAR + NBAR	31.27	0.0044	29.44	3.46	33.81	28.44	0.19
DSR + PAR + NDVI + EVI + LAI + FPAR + NBAR	31.21	0.0058	28.53	3.74	35.94	28.32	0.17

The first set of three trials was made with the R_n taken from the ground-based measurements. The second set of trials tested the effects of using DSR or PAR or both in combination with NDVI and EVI. Using both radiation variables with NDVI produced better results than using either of them separately. Using all four variables produced the lowest RMSEs at little additional computational cost. For all subsequent trials, both DSR and PAR were included.

The third set of comparisons tested the use of LAI and FPAR as input variables. The fourth set of trials included albedo as one of the input data variables. When albedo was the only input other than DSR and PAR the highest RMSEs for any of the combinations of input variables resulted. Including NDVI, EVI, LAI, and FPAR along with albedo improved the results to be similar to, or slightly better than, the trial with those variables but without albedo.

The final set of trials included NBAR as an input andproduced the lowest RMSEs of any combination.

Based on the overall patterns in these results, further tuning of all algorithms was conducted using all of the remote sensing input data variables: DSR, PAR, NDVI, EVI, LAI, FPAR, albedo, and NBAR.

3. Tuning of ML algorithms

Each of the algorithms that ran sufficiently quickly to be iterated with each training data set was optimized with that data set. The overall minimum RMSE results for the validation and test data sets for all algorithms are shown in Table 5.Notable aspects of the tuning are then described below for each "family" of algorithms.

Table 5: RMSE in W/m^2 for each optimized ML algorithm

	Small training data set		Large training data set	
Algorithm	Validation	Test	Validation	Test
RLR	30.55	29.84	31.22	30.23
LASSO	30.55	29.84	31.22	30.23
ELASTIC	30.55	29.84	31.22	30.23

KRR	23.85	23.41	Prohibitive	
RKS	25.35	25.52	22.22	22.10
TREE	29.19	28.71	25.14	25.45
BAGTREE	24.50	23.91	19.91	20.15
BOOST	28.86	28.33	28.21	27.86
NN, 1 HL	26.42	26.78	23.18	23.48
NN, 2 HL	25.76	25.20	21.58	22.69
NN, 3 HL	25.59	25.51	20.94	21.79
SVR	24.13	23.63	Prohibitive	

Note: Algorithms that are too computationally demanding for training with the large data set are labeled “Prohibitive”.

a. Linear regression variants

The linear regression variants demonstrated the weakest performance of all algorithm types. All linear regression variants (RLR, LASSO, and Elastic Net) show a pattern of optimum performance at zero or low regularization parameter values (<100 for RLR, <0.01 for LASSO), then worse performance or failure to converge as regularization parameters increase. Including the regularization parameters provided no advantage over a standard linear regression.

b. Kernel methods

When optimized, the kernel ridge regression performed better than any of the other algorithms with the small training data set, but it was too computationally demanding for use with the larger training data set. The RKS, which is in a sense a faster approximation of the KRR, did not perform as well with the smaller data set, but improved on that performance significantly with the larger training data set. Both KRR and RKS had more sensitivity to the kernel width parameter than to regularization, except for the RKS using a high number (> 1000) of random functions. Increasing the number of random functions usually produces better performance when optimized, but at the expense of more sensitivity to the other algorithm parameters.

c. Tree methods

The simple tree method was not sensitive to degree of pruning or number of data points required per partition. Therefore, these parameters were not adjusted in the trials with more complex tree algorithms.

Performance of the boosting tree method improved with increasing number of trees up to about 500 trees, then saturated with the larger data set and showed evidence of overfitting with the smaller data set with higher numbers of trees. (Figure 1). Boost tree algorithm performance was generally weak overall. The bagging tree algorithm was the strongest performer out of all of the algorithms with the large data set and shows improved performance with increasing number of trees and fraction of input data used to construct each, although a saturation effect is evident when the number of trees exceeds 100 (Figure 2).

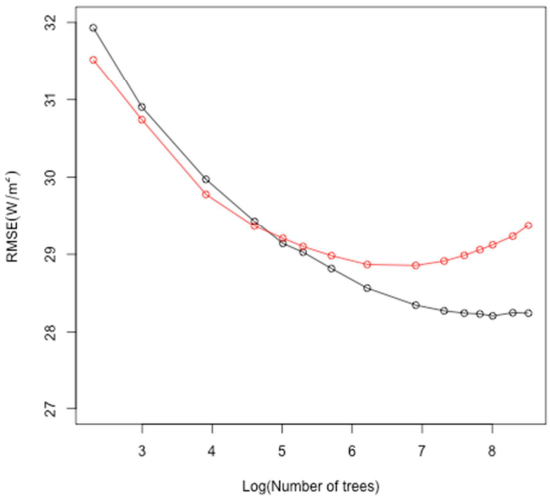

Figure 1: Validation RMSE versus number of trees used in boost tree algorithm. Red: Small training data set. Black: Large training data set.

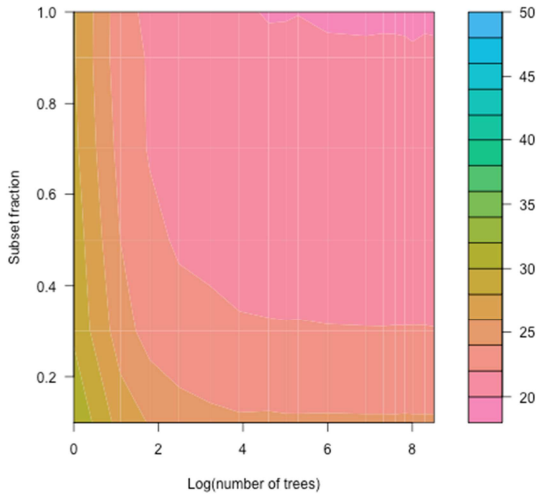

Figure 2: Validation RMSE versus number of trees and fraction of data included in each bagging tree using large training data set

d. Neural networks

The most notable result of the neural network trials is that two and three hidden layer networks outperform the single-layer neural network, especially for the larger data set. Performance generally improves with number of neurons in each layer up to about 50 to 100 neurons in the first layer but is less sensitive to the number of neurons in the second or third layer if they are present. Some evidence of overfitting is also present in all neural network results, since RMSE with the test data set exceeds that with the validation data set, by about 1 W/m^2 in the case of the 2 and 3 hidden layer NNs.

e. Support vector regression

The support vector regression method performed only modestly well with the smaller training data set, and tuning was computationally prohibitive with the larger training data set.

252

253 4. Trials with different ecosystem classes

254 The two and three hidden layer NN, RKS, and BAGTREE algorithms were used with the test
255 data sets for each of seven ecosystem types. Initially, the algorithms were optimized using
256 training and validation data from all sites. Then, each algorithm was tuned using the training and
257 validation data sets for each ecosystem type, then tested using the test data set for the same type.
258 Results with the full training data set are shown in Table 6, and results with the like-type-only
259 training data sets are given in Table 7.

260

261 Table 6: RMSE, R^2, and bias for different ecosystem types when ML algorithms are trained with
262 data from all sites.

Agricultural	**RMSE**(W/m^2)	**R^2**	**Bias**(W/m^2)
2 hidden layer NN	32.3557	0.6680	-0.6399
3 hidden layer NN	23.9863	0.8035	-1.0610
RKS	26.2128	0.7637	-1.1795
BAGTREE	17.8557	0.8950	-0.8671
Deciduous			
2 hidden layer NN	20.2389	0.7416	3.4893
3 hidden layer NN	18.9362	0.7741	2.0059
RKS	20.3107	0.7399	4.2604
BAGTREE	13.4676	0.8918	3.2371
Evergreen			
2 hidden layer NN	19.0075	0.6874	-0.5384

3 hidden layer NN	18.0615	0.7186	-0.4556
RKS	19.3665	0.6751	-0.0450
BAGTREE	12.4909	0.8745	-0.0113
Grassland			
2 hidden layer NN	18.4500	0.7853	-0.1038
3 hidden layer NN	17.9347	0.7981	-0.1971
RKS	18.7163	0.7791	-0.2180
BAGTREE	12.0934	0.9140	-0.0882
Savannah			
2 hidden layer NN	16.1591	0.8025	-0.8624
3 hidden layer NN	14.7330	0.8351	-0.5861
RKS	16.3950	0.8006	-1.3287
BAGTREE	16.3950	0.8006	-1.3287
Shrub			
2 hidden layer NN	33.8698	0.3823	-0.3441
3 hidden layer NN	16.6110	0.7777	-0.1439
RKS	18.0345	0.7395	-0.8539
BAGTREE	11.4718	0.9025	-0.6610
Wetland			
2 hidden layer NN	29.5516	0.8038	-1.3708
3 hidden layer NN	28.9601	0.8122	-1.8941
RKS	35.2762	0.7212	-4.4138

BAGTREE	22.9066	0.8878	-4.3023

263

264

265

266 Table 7: RMSE, R^2, and bias for different ecosystem types with ML algorithms optimized with

267 training and validation data from the same ecosystem type.

Agricultural	**RMSE** (W/m^2)	**R^2**	**Bias** (W/m^2)
2 hidden layer NN	27.7259	0.7354	-0.3191
3 hidden layer NN	29.6665	0.6971	-1.3922
RKS	27.2730	0.7440	-0.5729
BAGTREE	18.5799	0.8851	-0.7525
Deciduous			
2 hidden layer NN	28.3526	0.4983	-2.0546
3 hidden layer NN	33.7258	0.2953	-15.1065
RKS	28.7253	0.5318	6.5990
BAGTREE	25.8699	0.5878	0.1247
Evergreen			
2 hidden layer NN	24.2847	0.4945	-7.8675
3 hidden layer NN	24.8439	0.4796	-6.3154
RKS	23.6216	0.5208	-6.5283
BAGTREE	23.7131	0.5169	-3.6958
Grassland			

2 hidden layer NN	19.6553	0.7563	-0.3862
3 hidden layer NN	32.7546	0.3608	-0.3290
RKS	19.7727	0.7533	0.0463
BAGTREE	14.6462	0.8689	0.5006
Savannah			
2 hidden layer NN	16.9383	0.7830	0.6077
3 hidden layer NN	15.1067	0.8267	-0.2368
RKS	15.3770	0.8203	-0.3738
BAGTREE	12.9098	0.8779	-0.3341
Shrub			
2 hidden layer NN	17.0708	0.7674	0.5069
3 hidden layer NN	17.9521	0.7403	0.7350
RKS	17.5338	0.7522	0.0286
BAGTREE	12.0077	0.8851	0.3511
Wetland			
2 hidden layer NN	29.4605	0.8055	-0.3557
3 hidden layer NN	29.4891	0.8055	-0.3742
RKS	48.4014	0.4812	-0.1796
BAGTREE	24.9083	0.8606	1.5604

The results shown in Tables 6 and 7 show that the ML algorithms performed best for evergreen, grassland, and shrub sites. Performance was usually worst for wetland sites. The BAGTREE algorithm was the best performer in most cases, except for the savannah sites when the

algorithms were trained with all data and the evergreen sites when training was done with data from the same site type only. Training with data of the same type led to improved algorithm performance only in the case of savannah sites. This is probably related to the fact that the optimized algorithm parameters for the smaller individual site type training data sets reached less complexity (fewer neurons in neural networks, fewer random functions in RKS, and fewer trees used by BAGTREE) before overfitting became an issue than for the larger all site training data set.

Discussion

Here we systematically compared several machine learning methods for obtaining LE from a smaller or larger remote sensing only input data set. The best results for the smalltraining data set were with the kernel ridge regression (KRR), which was not viable with the large training set. Three of the other algorithms (RKS, BAGTREE, and multi-layer neural networks) produced a lower RMSE with the large training data set than the lowest RMSE attained with the small training data set. The cloud-detection example given in Pérez-Suay et al. (2017) also demonstrated this dynamic between the KRR and RKS methods. Here we also had good performance with the RKS, but even better performance with the bagging tree and multi-layer neural network.

Other than weaker performance by the linear regression variants, no family of methodsoutperformed the rest. Regularization of the linear regression variants did not produce any improvement to the algorithm results over a standard linear regression.

It has been shown that, while some of the ML algorithms perform well in terms of both accuracy and computational demand, there is also some tradeoff between training efficiency and performance. This is seen most clearly in the results with the large training data set, where the BAGTREE algorithm produced the lowest RMSE but required more run time than the RKS, boost tree, or smaller neural networks.The RKS algorithm is appealing due to its computational efficiency and low test RMSE. Increasing the number of random functions in the RKS generally reduces the optimized error, but also renders the algorithm more sensitive to its other parameters. It is notable that the "deeper" 2- and 3- layer neural networks tested in this study performed better than the single-layer neural network, since most studies in which neural networks are applied to the LE problem only make use of single hidden layer neural networks. The multi-layer neural networks only performed at their best if there were at least 50 neurons in the first layer but showed less sensitivity to the numbers of neurons in deeper layers. The neural networks showed more evidence of overfitting than any of the other algorithms, although the difference between validation and test data set RMSEs was only about 1 W/m^2.

Comparison of ML algorithm performance when trained with data from individual ecosystem types instead of data from all sites usually showed worse performance, except for savannah sites. This contrasts with the modest improvement found by Carter and Liang (2018) when non-ML LE algorithms were tuned using data from individual ecosystem types. However, the poor performance of the algorithms for wetland sites is consistent with Carter and Liang (2018). It appears that the ability of the ML algorithms to extract more complex patterns from larger data sets usually outweighs any advantage gained by restricting training data to one site type only.

Conclusions

A comparison of ten ML methods for obtaining LE from a combination of remote sensing data (GLASS and MODIS) was performed in terms of accuracy and speed. The results showed wide variation in algorithm efficiency.Including more input variables improved the results with little or no additional computational cost. Use of GLASS radiation products produced results comparable to using ground-based net radiation measurements. Inclusion of NBAR as one of the parameters produced the best results.

The best performance with a smaller training data set was obtained using the kernel ridge regression (KRR), which was too computationally demanding for use with the larger data set. The best performance with the larger data set was achieved by the bootstrap aggregation tree (BAGTREE) method, followed by the random kernel (RKS) and multiple hidden layer neural network (NN) methods. The BAGTREE, neural network, and RKS algorithm performance could be improved modestly for some ecosystem types by using training data from that ecosystem type only.

Since the machine learning techniques evaluated here can be applied to any combination of input variables, it should be possible to use them to generate global, long-term records of LE. The GLASS data sets (Liang et al. 2013, 2014), which include albedo (Qu et al. 2014; Liu et al. 2013), leaf area index (Xiao et al. 2016, 2017a), and NDVI (Xiao et al. 2017b) in addition to radiation variables, are based on the AVHRR and MODIS records, and therefore provide the opportunity to examine global LE trends over decades.

Acknowledgements: This work was partially supported by funding from the United States National Aeronautics and Space Administration (NASA) and National Oceanic and Atmospheric Administration (NOAA).Funding for AmeriFlux data resources was provided by the U.S. Department of Energy's Office of Science.

References:

Carter, Corinne, and Shunlin Liang. 2018. "Comprehensive Evaluation of Empirical Algorithms for Estimating Land Surface Evapotranspiration." *Agricultural and Forest Meteorology* 256–257: 334–45.

Deo, Ravinesh C., Pijush Samui, and Dookie Kim. 2016. "Estimation of Monthly Evaporative Loss Using Relevance Vector Machine, Extreme Learning Machine and Multivariate Adaptive Regression Spline Models." *Stochastic Environmental Research and Risk Assessment* 30 (6): 1769–84. https://doi.org/10.1007/s00477-015-1153-y.

Didan, K. 2015. "MOD13A1 MODIS/Terra Vegetation Indices 16-Day L3 Global 500m SIN Grid V006." NASA EOSDIS Land Processes DAAC. https://doi.org/10.5067/MODIS/MOD13A1.006.

Dou, Xianming, and Yongguo Yang. 2018. "Evapotranspiration Estimation Using Four Different Machine Learning Approaches in Different Terrestrial Ecosystems." *Computers and Electronics in Agriculture* 148 (May): 95–106. https://doi.org/10.1016/j.compag.2018.03.010.

Hastie, Trevor, Robert Tibshirani, and Jerome Friedman. 2009. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer Science + Business Media, LLC.

Helman, D., A. Givati, and I. M. Lensky. 2015. "Annual Evapotranspiration Retrieved from Satellite Vegetation Indices for the Eastern Mediterranean at 250 m Spatial Resolution." *Atmospheric Chemistry and Physics* 15 (21): 12567–79. https://doi.org/10.5194/acp-15-12567-2015.

Jung, Martin, Markus Reichstein, Philippe Ciais, Sonia I. Seneviratne, Justin Sheffield, Michael L. Goulden, Gordon Bonan, et al. 2010. "Recent Decline in the Global Land Evapotranspiration Trend Due to Limited Moisture Supply." *Nature* 467 (7318): 951–54. https://doi.org/10.1038/nature09396.

Kaheil, Y.H., E. Rosero, M.K. Gill, M. McKee, and L.A. Bastidas. 2008. "Downscaling and Forecasting of Evapotranspiration Using a Synthetic Model of Wavelets and Support Vector Machines." *IEEE Transactions on Geoscience and Remote Sensing* 46 (9): 2692–2707. https://doi.org/10.1109/TGRS.2008.919819.

Kalma, Jetse D., Tim R. McVicar, and Matthew F. McCabe. 2008. "Estimating Land Surface Evaporation: A Review of Methods Using Remotely Sensed Surface Temperature Data." *Surveys in Geophysics* 29 (4–5): 421–69. https://doi.org/10.1007/s10712-008-9037-z.

Ke, Yinghai, Jungho Im, Seonyoung Park, and Huili Gong. 2016. “Downscaling of MODIS One Kilometer Evapotranspiration Using Landsat-8 Data and Machine Learning Approaches.” *Remote Sensing* 8 (3): 215. https://doi.org/10.3390/rs8030215.

———. 2017. “Spatiotemporal Downscaling Approaches for Monitoring 8-Day 30 m Actual Evapotranspiration.” *ISPRS Journal of Photogrammetry and Remote Sensing* 126 (April): 79–93. https://doi.org/10.1016/j.isprsjprs.2017.02.006.

Lazaro-Gredilla, Miguel, and Michalis K. Titsias. 2011. “Variation Heteroschedastic Gaussian Process Regression.” *28th International Conference on Machine Learning (ICML 2011)*.

Lazaro-Gredilla, Miguel, Michalis K. Titsias, Jochem Verrelst, and Gustavo Camps-Valls. 2014. “Retrieval of Biophysical Parameters With Heteroscedastic Gaussian Processes.” *IEEE Geoscience and Remote Sensing Letters* 11 (4): 838–42. https://doi.org/10.1109/LGRS.2013.2279695.

Liang, Shunlin, Xiaotong Zhang, Zhiqiang Xiao, Jie Cheng, Qiang Liu, and Xiang Zhao. 2014. *Global LAnd Surface Satellite (GLASS) Products Algorithms, Validation and Analysis*.

Liang, Shunlin, Xiang Zhao, Suhong Liu, Wenping Yuan, Xiao Cheng, Zhiqiang Xiao, Xiaotong Zhang, et al. 2013. “A Long-Term Global LAnd Surface Satellite (GLASS) Data-Set for Environmental Studies.” *International Journal of Digital Earth* 6 (sup1): 5–33. https://doi.org/10.1080/17538947.2013.805262.

Liu, Qiang, Lizhao Wang, Ying Qu, Nanfeng Liu, Suhong Liu, Hairong Tang, and Shunlin Liang. 2013. “Preliminary Evaluation of the Long-Term GLASS Albedo Product.” *International Journal of Digital Earth* 6 (sup1): 69–95. https://doi.org/10.1080/17538947.2013.804601.

Murphy, Kevin P. 2012. *Machine Learning: A Probabilistic Perspective*. MIT Press.

Myneni, R., Y. Knyazikhin. 2015. “MOD15A2H MODIS/Terra Leaf Area Index/FPAR 8-Day L4 Global 500m SIN Grid V006.” NASA EOSDIS Land Processes DAAC. https://doi.org/10.5067/MODIS/MOD15A2H.006.

Park, Seonyoung, Jungho Im, Eunna Jang, and Jinyoung Rhee. 2016. “Drought Assessment and Monitoring through Blending of Multi-Sensor Indices Using Machine Learning Approaches for Different Climate Regions.” *Agricultural and Forest Meteorology* 216 (January): 157–69. https://doi.org/10.1016/j.agrformet.2015.10.011.

Pérez-Suay, Adrián, Julia Amorós-López, Luis Gómez-Chova, Valero Laparra, Jordi Muñoz-Marí, and Gustau Camps-Valls. 2017. “Randomized Kernels for Large Scale Earth Observation Applications.” *Remote Sensing of Environment* 202 (December): 54–63. https://doi.org/10.1016/j.rse.2017.02.009.

Qu, Ying, Qiang Liu, Shunlin Liang, Lizhao Wang, Nanfeng Liu, and Suhong Liu. 2014. “Direct-Estimation Algorithm for Mapping Daily Land-Surface Broadband Albedo From MODIS Data.” *IEEE Transactions on Geoscience and Remote Sensing* 52 (2): 907–19. https://doi.org/10.1109/TGRS.2013.2245670.

Rhee, Jinyoung, and Jungho Im. 2017. “Meteorological Drought Forecasting for Ungauged Areas Based on Machine Learning: Using Long-Range Climate Forecast and Remote Sensing Data.” *Agricultural and Forest Meteorology* 237–238 (May): 105–22. https://doi.org/10.1016/j.agrformet.2017.02.011.

Schaaf, C. Z. Wang. 2015a. “MCD43A3 MODIS/Terra+Aqua BRDF/Albedo Daily L3 Global - 500m V006.” NASA EOSDIS Land Processes DAAC. https://doi.org/10.5067/MODIS/MCD43A3.006.

———. 2015b. “MCD43A4 MODIS/Terra+Aqua BRDF/Albedo Nadir BRDF Adjusted RefDaily L3 Global - 500m V006.” NASA EOSDIS Land Processes DAAC. https://doi.org/10.5067/MODIS/MCD43A4.006.

Smola, A. J., and B. Scholkopf. n.d. “A Tutorial on Support Vector Regression.” *Statistics and Computing* 14 (3): 199–222.

Thayananthan, A, R Navaratnam, B. Stenger, P. H. S. Torr, and R Cipolla. 2006. “Multivariate Relevance Vector Machines for Tracking.” Edited by A Leonardis and A Pinz. *Computer Vision- ECCV 2006, Pt 3, Proceedings*, Lecture Notes in Computer Science, 3953: 124–38.

Wang, Kaicun, and Robert E. Dickinson. 2012. “A Review of Global Terrestrial Evapotranspiration: Observation, Modeling, Climatology, and Climatic Variability: GLOBAL TERRESTRIAL EVAPOTRANSPIRATION.” *Reviews of Geophysics* 50 (2). https://doi.org/10.1029/2011RG000373.

Wang, Kaicun, and Shunlin Liang. 2008. “An Improved Method for Estimating Global Evapotranspiration Based on Satellite Determination of Surface Net Radiation, Vegetation Index, Temperature, and Soil Moisture.” *Journal of Hydrometeorology* 9 (4): 712–27. https://doi.org/10.1175/2007JHM911.1.

Xiao, Zhiqiang, Shunlin Liang, and Bo Jiang. 2017. “Evaluation of Four Long Time-Series Global Leaf Area Index Products.” *Agricultural and Forest Meteorology* 246 (November): 218–30. https://doi.org/10.1016/j.agrformet.2017.06.016.

Xiao, Zhiqiang, Shunlin Liang, Xiaodan Tian, Kun Jia, Yunjun Yao, and Bo Jiang. 2017. “Reconstruction of Long-Term Temporally Continuous NDVI and Surface Reflectance From AVHRR Data.” *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 10 (12): 5551–68. https://doi.org/10.1109/JSTARS.2017.2744979.

Xiao, Zhiqiang, Shunlin Liang, Jindi Wang, Yang Xiang, Xiang Zhao, and Jinling Song. 2016. “Long-Time-Series Global Land Surface Satellite Leaf Area Index Product Derived From MODIS and AVHRR Surface Reflectance.” *IEEE Transactions on Geoscience and Remote Sensing* 54 (9): 5301–18. https://doi.org/10.1109/TGRS.2016.2560522.

Yao, Yunjun, Shunlin Liang, Jie Cheng, Shaomin Liu, Joshua B. Fisher, Xudong Zhang, Kun Jia, et al. 2013. “MODIS-Driven Estimation of Terrestrial Latent Heat Flux in China Based on a Modified Priestley?Taylor Algorithm.” *Agricultural and Forest Meteorology* 171–172 (April): 187–202. https://doi.org/10.1016/j.agrformet.2012.11.016.

Yao, Yunjun, Shunlin Liang, Xianglan Li, Jiquan Chen, Kaicun Wang, Kun Jia, Jie Cheng, et al. 2015. “A Satellite-Based Hybrid Algorithm to Determine the Priestley–Taylor Parameter for Global Terrestrial Latent Heat Flux Estimation across Multiple Biomes.” *Remote Sensing of Environment* 165 (August): 216–33. https://doi.org/10.1016/j.rse.2015.05.013.

Yao, Yunjun, Shunlin Liang, Qiming Qin, Kaicun Wang, and Shaohua Zhao. 2011. “Monitoring Global Land Surface Drought Based on a Hybrid Evapotranspiration Model.” *International Journal of Applied Earth Observation and Geoinformation* 13 (3): 447–57. https://doi.org/10.1016/j.jag.2010.09.009.

Yebra, Marta, Albert Van Dijk, Ray Leuning, Alfredo Huete, and Juan Pablo Guerschman. 2013. “Evaluation of Optical Remote Sensing to Estimate Actual Evapotranspiration and Canopy Conductance.” *Remote Sensing of Environment* 129 (February): 250–61. https://doi.org/10.1016/j.rse.2012.11.004.

Zhang, Ke, John S. Kimball, and Steven W. Running. 2016. “A Review of Remote Sensing Based Actual Evapotranspiration Estimation: A Review of Remote Sensing

Evapotranspiration.” *Wiley Interdisciplinary Reviews: Water* 3 (6): 834–53. https://doi.org/10.1002/wat2.1168.

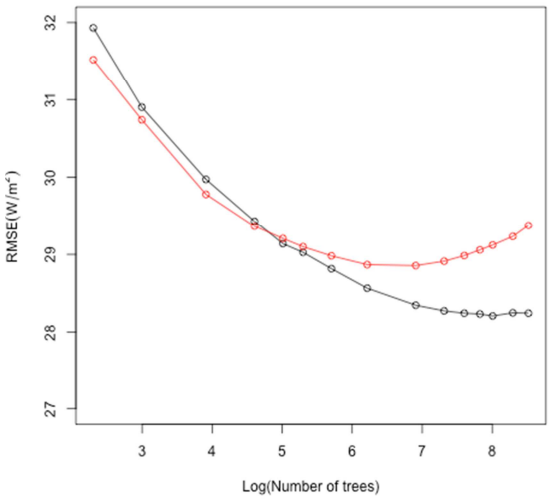

Figure 1: Validation RMSE versus number of trees used in boost tree algorithm. Red: Small training data set. Black: Large training data set.

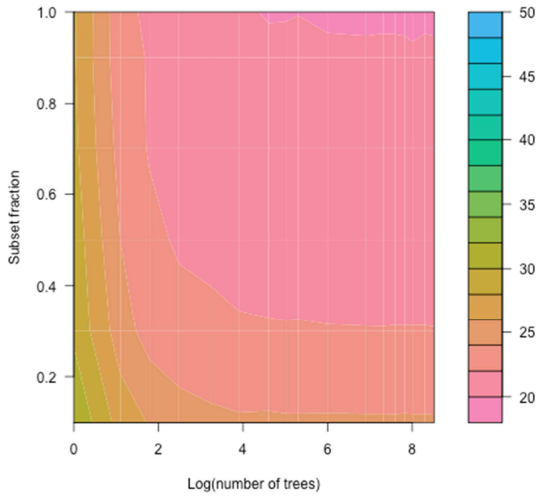

Figure 2: Validation RMSE versus number of trees and fraction of data included in each bagging tree using large training data set