To appear in the *International Journal of Remote Sensing*
Vol. 00, No. 00, Month 20XX, 1–22

Automatic Boosted Flood Mapping from Satellite Data

Brian Coltin*, Scott McMichael, Trey Smith, and Terrence Fong

Intelligent Robotics Group, NASA Ames, Moffett Field, CA

(*Received 00 Month 20XX; accepted 00 Month 20XX*)

Numerous algorithms have been proposed to map floods from Moderate Resolution Imaging Spectroradiometer (MODIS) imagery. However, most require human input to succeed, either to specify a threshold value or to manually annotate training data. We introduce a new algorithm based on Adaboost which effectively maps floods without any human input, allowing for a truly rapid and automatic response. The Adaboost algorithm combines multiple thresholds to achieve results comparable to state-of-the-art algorithms which do require human input. We evaluate Adaboost, as well as numerous previously proposed flood mapping algorithms, on multiple MODIS flood images, as well as on hundreds of non-flood MODIS lake images, demonstrating its effectiveness across a wide variety of conditions.

Keywords: floods; flood mapping; MODIS; Adaboost

1. Introduction

Every year, floods claim an average of 140 lives and cause $6 billion in property damage in the United States alone (United States Geological Survey 2006). With maps of the flooded areas, produced rapidly and automatically from satelite or aerial imagery, responders can allocate their resources effectively to minimize loss of life and damage by quickly identifying the flooded areas (Taubenböck et al. 2011).

Our ultimate goal, working with our partners at Google, is to deploy an online tool which will automatically and rapidly create and publish maps of floods. These maps will be provided both to responders and to flood victims, complementing Google's existing crisis response efforts.

The remote sensing community has already developed numerous algorithms and online tools which rapidly map the extent of floods using multiple sources of satellite imagery. Existing online tools include the Dartmouth Flood Observatory (Brakenridge and Anderson 2006), the Global Flood Detection System (Kugler and De Groeve 2007), and the Global Flood Monitoring System (Wu et al. 2014). However, these tools focus largely on the needs of researchers rather than the needs of flood responders and victims, as they are in a format not readily accessible or understandable by laypeople.

While preparing our own online flood mapping tool, we discovered that, despite substantial previous research in flood mapping algorithms, comparative, quantitative evaluation has been scant at best. Many algorithms have been introduced

[*] Corresponding author. Email: brian.j.coltin@nasa.gov

for specific floods or regions, and their effectiveness has only been verified by subjective visual evaluation. Without quantitative evaluation, it is difficult to know which algorithm to use in a given environment. To the best of our knowledge, only one quantitative evaluation of water detection algorithms has been done (Boschetti et al. 2014), but its focus was on flooded rice cropping systems, not floods. In addition, the only algorithms evaluated were normalized difference spectral indices thresholds. The lack of previous quantitative evaluations is largely due to three factors:

(1) The challenge of establishing the *ground truth* flooding conditions with certainty. Aside from taking physical measurements at the site of the flood, there is no good way to establish the true flooding boundaries aside from evaluation of the geospatial imagery.
(2) The difficulty of acquiring *imagery for multiple flood locations*. Data sources like the Moderate Resolution Imaging Spectroradiometer (MODIS) and Landsat have large areas and times of coverage but are often occluded by clouds, while little SAR flood data is freely available. Furthermore, many researchers develop algorithms with the intent of studying only a single area.
(3) The *computational challenges* inherent in processing the imagery. Acquiring imagery is expensive in both time and bandwidth. Furthermore, extensive computational resources are required to process the imagery. However, processing images on a global scale is becoming more and more feasible as computational capabilities improve (Klein et al. 2015).

In this article we survey, compare and evaluate multiple flood mapping algorithms with the intent of choosing an algorithm for an online, fully automatic flood mapping tool. For challenge *(1)*, establishing ground truth, we evaluate the algorithms against our best-guess manually generated expert flood maps, as well as in hundreds of non-flood conditions using the permanent water mask. We address challenges *(2)* and *(3)* with the use of Google Earth Engine, an online platform for performing massively parallel computation on geospatial imagery. Earth Engine simplifies the research by pre-loading all of the imagery for common sources such as MODIS and Landsat, and enabling simple and fast parallel processing of the imagery using Google's servers. Earth Engine has enabled us to evaluate multiple algorithms on hundreds of images. However, Earth Engine is limited to certain types of operations, mainly those which are easily parallelizable or local per pixel computations. Hence, our implementation of some algorithms differs slightly from the literature version, and we clearly note these changes. Our algorithms and data are released as open source software so that future researchers can benefit from our work (available at https://github.com/nasa/CrisisMappingToolkit).

From the results of the evaluation, we realized that no existing algorithm succeeds in every scenario, and that the most successful existing algorithms require human input for tuning or training. For our intended fully automatic deployment, triggered by online flood alerts, any human input is unacceptable. Hence, we propose the use of Adaboost for flood mapping. Adaboost combines multiple weaker classifiers to make an algorithm that is much more robust to varying conditions than any one classifier on its own. The key insight which enables mapping without human input is that non-flood conditions can effectively be used for training the weak classifiers.

We focus our flood mapping efforts on data from the Moderate Resolution Imaging Spectroradiometer (MODIS). MODIS is a sensor aboard the Terra and Aqua satelites which observes 36 spectral bands. MODIS observes the Earth's entire surface every one to two days, making it effective for rapid flood mapping. However, it cannot see through clouds and the bands are observed with low ground resolution of 250 m/pixel or 500 m/pixel (Barnes, Pagano, and Salomonson 1998).

Several other data sources have also been used to map floods, including Synethic Aperture Radar (SAR) (Martinis 2010; Matgen et al. 2011; Martinis, Twele, and Voigt 2009), microwave data from the Advanced Microwave Scanning Radiometer - Earth Observing System (ASMR-E) (Kugler, De Groeve, and Thierry 2007), precipitation estimates from the Tropical Rainfall Measuring Mission (TRMM) (Wu et al. 2014), and data from the Advanced Along-Track Scanning Radiometer (Fichtelmann and Borg 2012). Other researchers have also developed predictive flood inundation maps in preparation for future flood events. For example, the Global Flood Awareness System predicts floods in advance using weather predictions and a hydrological model (Alfieri et al. 2013). The United States Geological Survey (USGS) creates libraries of inundation maps for numerous locations which describe the expected flood extent given the readings on nearby USGS stream gauges (United States Geological Survey 2015). Furthermore, remote sensing of water is useful not only for the formation of wet / dry maps, but also to develop hydraulic models to better understand and mitigate flood damage (Schumann et al. 2009).

In the remainder of this article, we first present existing flood mapping algorithms for MODIS data. Next, we introduce our Adaboost algorithm, which effectively and fully automatically classifies floods. Finally, we present an extensive evaluation of the effectiveness of our approach on a small number of floods and on hundreds of lakes.

2. Review of MODIS Flood Mapping Algorithms

The two MODIS satellites, Terra and Aqua, achieve global coverage every one to two days which makes them suitable for rapid response flood mapping. They measure 36 spectral bands. We denote the surface reflectance from MODIS band i as B_i. The wavelengths measured by the MODIS bands are: B_1, 620-670 nm; B_2, 841-876 nm; B_3, 459-479 nm; B_4, 545-565 nm; B_5, 1230-1250 nm; B_6, 1628-1652 nm; and B_7, 2105-2155 nm. The remaining bands are unused in this work. B_1 and B_2 have a 250m resolution, B_3–B_7 have a 500m resolution, and the remaining bands have a 1000m resolution (Barnes, Pagano, and Salomonson 1998). This low resolution may not be sufficient to determine if individual homes are flooded, but is useful in directing large scale response efforts.

Another drawback of MODIS is that it is often occluded by clouds. Numerous cloud detection algorithms have been developed for MODIS (Frey et al. 2008), and it is possible to filter clouds before applying any flood detection algorithms. However, to better focus our evaluation on flood detection algorithms, we evaluated cloud-free MODIS scenes only and set aside the challenge of cloud filtering.

We first review a number of algorithms from the literature for flood mapping with MODIS, which we later evaluate using Google Earth Engine. Our evaluation is not exhaustive, and some notable omissions include transforming MODIS images to the Hue Saturation Value (HSV) color space for flood detection (Pekel et al. 2014) and dynamic thresholding based on tiling (Klein et al. 2015).

Table 1. Table of MODIS indices.

Index	Equation
Normalized Difference Water Index (NDWI)	$\frac{B_1-B_6}{B_1+B_6}$
Land Surface Water Index (LSWI)	$\frac{B_2-B_6}{B_2+B_6}$
Normalized Difference Vegetation Index (NDVI)	$\frac{B_2-B_1}{B_2+B_1}$
Enhanced Vegetation Index (EVI)	$\frac{2.5(B_2-B_1)}{6B_1+B_2-7.5B_3+1}$

2.1. *Thresholding Algorithms*

One of the simplest and most common flood mapping algorithms for MODIS data is to apply thresholds to one or more indices computed from the higher resolution MODIS bands. These thresholding algorithms can achieve good results, are computationally inexpensive, and are easily implemented. However, they must often be calibrated for a specific region or dataset, as the most discriminating thresholds are highly dependent on both the water content and the surrounding land's spectral properties. This need for human input to select a threshold precludes fully automatic deployment.

2.1.1. Islam

The Islam algorithm thresholds an image based on the Enhanced Vegetation Index (EVI) and Land Surface Water Index ($LSWI$) (see Table 1 for definitions of MODIS indices). Pixels which satisfy the formula

$$((\text{EVI}) \leq 0.3 \wedge (\text{EVI}) - (\text{LSWI}) \leq 0.05) \vee ((\text{EVI}) \leq 0.05 \wedge (\text{LSWI}) \leq 0.0) \quad (1)$$

are marked as water (where $\wedge$ indicates logical and, and $\vee$ is logical or). The original algorithm further segments these pixels into flooded, partially flooded, and permanent water bodies (Islam, Bala, and Haque 2009). The Islam algorithm does not contain any adjustable thresholds, and was originally developed solely for a flood in Bangladesh; thus, it was uncertain how and if it would generalize to floods in other regions. This does give it the advantage of being fully automatic, not requiring any human intervention.

2.1.2. Xiao

A similar approach used to detect floods is the Xiao algorithm. Pixels which satisfy the formula

$$((\text{LSWI}) - (\text{EVI}) \geq 0.05) \vee (2(\text{LSWI}) - (\text{NDVI}) \geq 0.05) \quad (2)$$

are marked as flooded. This approach, together with a further filtering step, was originally designed to detect rice paddies and was successfully applied in southeast Asia (Xiao et al. 2006). Like Islam, the Xiao algorithm has the advantage of being human input-free.

2.1.3. Diff

The Diff algorithm classifies pixels which satisfy the formula

$$B_2 - B_1 \leq K_{\text{Diff}} \quad (3)$$

as flooded. The threshold K_{DIFF} is a parameter that is chosen for each region based on the properties of the water and surrounding land areas. This simple algorithm is surprisingly effective. However, it requires a human to specify the threshold manually.

2.1.4. Dart

The Dartmouth flood observatory maps worldwide surface water with MODIS using a cloud filter together with a thresholding algorithm (Brakenridge and Anderson 2006). The Dart algorithm classifies pixels which satisfy the equation

$$\frac{B_2 + C}{B_1 + D} \leq K_{\text{DART}} \tag{4}$$

as flooded. This algorithm requires three constants to be specified, C, D, and K_{DART}, all of which are determined empirically (Brakenridge 2012).

2.1.5. Mndwi

The study by Boschetti et al. (2014) compares a large number of normalized difference spectral indices (indices in the form $\frac{A-B}{A+B}$) and evaluates them on a few regions of rice crops. One of the most effective indices in this study was the Modified Normalized Difference Water Index:

$$\frac{B_6 - B_4}{B_6 + B_4} \leq K_{\text{MNDWI}}. \tag{5}$$

We evaluate the Mndwi as representative of normalized difference spectral indices. The constant K_{MNDWI} must be manually specified for each problem instance.

2.1.6. Fai

The Floating Algae Index (Fai) classifies pixels where

$$B_2 - \left(B_1 + \frac{859 - 645}{1240 - 645}\left(B_5 - B_1\right)\right) \leq K_{\text{FAI}} \tag{6}$$

as flooded. The Fai is promising for separating land and water because it is believed to be less sensitive than other indices to local environmental conditions (Feng et al. 2012). The constant K_{FAI} must be manually specified for each problem instance.

2.1.7. Theme

The Thematic MODIS Processor (Theme) is a decision tree using EVI/LSWI thresholds, slope measurements, and region growing steps to classify pixels into one of six flood-related output classes. It is the most complicated of the thresholding approaches we considered; for a full description see Martinis et al. (2013).

First, Theme begins with the water classification from the Islam algorithm. Then, flooded pixels on which the DEM shows a high slope are marked as unflooded. Finally, a region growing step marks pixels neighboring flooded pixels as flooded if they satisfy a looser version of the Islam thresholding constraints. Note that due to Earth Engine limitations, we approximate the region growing step.

2.2. *Supervised Learning*

The next class of algorithms we consider are supervised learning approaches. Where the thresholding approaches require a human operator to provide at most a single threshold value, supervised learning approaches require training data in the form of an annotated flood region. The algorithm uses this region to learn how to identify floods in additional data. For supervised learning algorithms to be successful the training data must be similar to the test data.

Supervised learning algorithms are expected to outperform the other algorithms (depending on the quality of the training data) as they have more data to work with and draw conclusions from, but the requirement of human-produced training data renders them less promising for rapid response. It is possible that with enough data, a general classifier could be learned for any flood. Unfortunately, due to the rarity of flood events and the huge variation among ground conditions this is challenging.

A number of features have previously been suggested for supervised flood mapping: B_1, B_2, $B_2 - B_1$, $\frac{B_2}{B_1}$, the NDWI, and the NDVI (Sun, Yu, and Goldberg 2011). Using these features, we consider three different supervised learning approaches:

- **Cart**: A classification and regression tree (Olshen and Stone 1984) forms a tree of rules by which to classify flooded and unflooded pixels. Cart was previously proposed for use in flood mapping (Sun, Yu, and Goldberg 2011).
- **Rf**: The Random Forests algorithm constructs an ensemble of decision trees which then vote on whether a pixel is flooded (Breiman 2001). Rf was previously proposed for use in flood mapping (Sun, Yu, and Goldberg 2011).
- **Svm**: The final supervised learning approach is a support vector machine solved with the Pegasos solver (Shalev-Shwartz et al. 2011). While to our knowledge it has not previously been used for flood mapping, it is a popular machine learning algorithm.

2.3. *Dynamic Nearest Neighbor Searching*

The previous thresholding and supervised learning approaches (with the exception of Theme) all operate on the level of individual pixels, and do not take any contextual information into account. However, the surrounding pixels are an important source of additional information. Intuitively, when humans solve the flood mapping problem, they are completely unable to determine whether individual pixels in isolation are flooded. It is only when presented with groups of pixels that patterns emerge and humans are able to identify flooding. The two Dynamic Nearest Neighbor Searching (DNNS) algorithms rely heavily on measurements of nearby locations to classify each pixel.

2.3.1. Dnns

Dnns incorporates contextual information. As with the supervised learning algorithms, training data is required.

The Dnns algorithm (Li et al. 2013b) is shown in Algorithm 1. First, "pure water" and "pure land" pixels are determined from a variant of the Rf classifier which outputs "mixed" pixels as well. Then, for each pixel in the image, Dnns averages the surrounding pixels detected as pure water to compute an estimated regional water reflectance. From this pure water reflectance, the neighboring land pixels are determined, and an average land reflectance is estimated. From the average water and land reflectance and the MODIS B_6 value, a "water fraction", or how much

```
function Dnns($\mathbf{B}$)
    $\mathrm{W_{pure}} \leftarrow$ PureWater($\mathbf{B}$), $\mathrm{L_{pure}} \leftarrow$ PureLand($\mathbf{B}$)
    for $(i,j) \in \mathbf{B}$ do
        $\mathrm{N} \leftarrow \left\{(i',j') : \sqrt{(i-i')^2 + (j-j')^2} \leq R_\mathrm{D}\right\}$        ▷ Neighboring Pixels
        $W_\mathrm{mean} \leftarrow \frac{1}{|\mathrm{N} \cap \mathrm{W_{pure}}|} \sum_{(i',j') \in \mathrm{N} \cap \mathrm{W_{pure}}} \mathbf{B}(i',j')$        ▷ Mean Water Reflectance
        $\mathrm{N_{land}} \leftarrow \left\{ (i',j') \in \mathrm{N} : \begin{array}{l} \frac{\mathbf{B}_1(i,j) - W_{\mathrm{mean},1}}{\mathbf{B}_6(i,j)} < \frac{\mathbf{B}_1(i',j')}{\mathbf{B}_6(i',j')} < \frac{\mathbf{B}_1(i,j)}{\mathbf{B}_6(i,j)}, \\ \frac{\mathbf{B}_2(i,j) - W_{\mathrm{mean},2}}{\mathbf{B}_6(i,j)} < \frac{\mathbf{B}_2(i',j')}{\mathbf{B}_6(i',j')} < \frac{\mathbf{B}_2(i,j)}{\mathbf{B}_6(i,j)} \end{array} \right\}$
        $L_\mathrm{mean} \leftarrow \frac{1}{|\mathrm{N_{land}}|} \sum_{(i',j') \in \mathrm{N_{land}}} \mathbf{B}(i',j')$        ▷ Mean Land Reflectance
        $\mathbf{W}_\mathrm{f}(i,j) \leftarrow \begin{cases} 0 & \text{if } (i,j) \in \mathrm{W_{pure}} \\ 1 & \text{if } (i,j) \in \mathrm{L_{pure}} \\ \frac{L_{\mathrm{mean},6} - \mathbf{B}_6(i,j)}{L_{\mathrm{mean},6} - W_{\mathrm{mean},6}} & \text{otherwise} \end{cases}$        ▷ Water Fraction
    end for
    return $\mathbf{W}_\mathrm{f}$
end function
```

Algorithm 1: The Dynamic Nearest Neighbor Searching flood mapping algorithm (Li et al. 2013b). **B** is the multi-band MODIS image, $(i,j) \in \mathbf{B}$ are the pixel coordinates. Edge cases where $|\mathrm{N} \cap \mathrm{W_{pure}}| = 0$, $|\mathrm{N_{land}}| = 0$ are omitted.

of the pixel contains water, is estimated. The water fraction is only applied to the mixed pixels.

2.3.2. Ddem

The DNNS with Digital Elevation Maps (DEMs) algorithm, Ddem, improves on Dnns by modifying the result based on a DEM (Li et al. 2013a). While MODIS has 500m and 250m pixels, the worldwide SRTM 90m DEM (Farr et al. 2007) and the United States' National Elevation Dataset 10m DEM (for the 48 continental US states only) (Gesch et al. 2002) are much higher resolution, allowing the construction of a more detailed flood map. However, this detail does not come from empirical observation, but from modelling of the flow of water based on the terrain model.

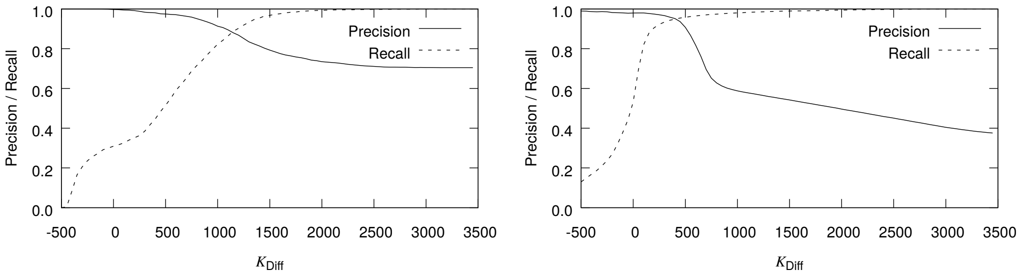
Ddem uses the water fraction from Dnns for each MODIS pixel and a histogram of the DEM elevations in the MODIS pixel to select the flood level consistent with the observed values. Then, since water levels are flat, a filtering step smooths the elevations of connected water bodies. The key idea is that the partially flooded pixels offer the most information about the water level. See Algorithm 2 for the ApplyDem algorithm, which takes the water fraction output from Dnns as input.

3. Fully Automatic Flood Mapping

Of the MODIS algorithms we reviewed, only Evi, Xiao, and Theme are fully automatic, requiring no human input. The other algorithms either require humans to specify a threshold value or provided annotated training data, making them unsuitable for rapid, fully automatic response. Furthermore, existing fully automatic algorithms fare poorly across different regions (as we will show in our experimental results). Hence, we introduce several algorithms which are more robust to differing

```
function ApplyDem($\mathbf{W}_\mathrm{f}$, $\mathbf{D}$)
    $\mathbf{H}, \mathbf{O} \leftarrow$ Empty Images
    for $(i, j) \in \mathbf{W}_\mathrm{f}$ do
        $W_\mathrm{min} \leftarrow \min_{\text{DEM pixels } (x,y) \text{ in } \mathbf{W}_\mathrm{f}(i,j)} \mathbf{D}(x, y)$
        $W_\mathrm{max} \leftarrow \max_{\text{DEM pixels } (x,y) \text{ in } \mathbf{W}_\mathrm{f}(i,j)} \mathbf{D}(x, y)$
        $\mathbf{H}(i, j) \leftarrow W_\mathrm{min} + \mathbf{W}_\mathrm{f}(i, j)\,(W_\mathrm{max} - W_\mathrm{min})$        ▷ Estimate Water Height
    end for
    for $(i, j) \in \mathbf{W}_\mathrm{f}$ do
        $\mathrm{W}_\mathrm{num} \leftarrow \left\{(i', j') : 0 < \mathbf{W}_\mathrm{f}(i', j') < 1 \wedge \sqrt{(i - i')^2 + (j - j')^2} \leq R_\mathrm{M}\right\}$
        $H_\mathrm{mean} \leftarrow \frac{1}{|\mathrm{W}_\mathrm{num}|} \sum_{(i',j') \in \mathrm{W}_\mathrm{num}} \mathbf{H}(i', j')$        ▷ Average Region Water Height
        for DEM pixels $(x, y) \in \mathbf{W}_\mathrm{f}(i, j)$ do
            $\mathbf{O}(x, y) \leftarrow \mathbf{D}(x, y) < H_\mathrm{mean}$        ▷ Fill DEM Under Water
        end for
    end for
    return $\mathbf{O}$
end function
```

Algorithm 2: The ApplyDem algorithm, which produces a flood map from a water fraction $\mathbf{W}_\mathrm{f}$ and a DEM $\mathbf{D}$. R_M is a constant, the radius within which to average water heights. Note that our water height estimation differs from the original algorithm (see Section 4.1.2).



(a) (b)

Figure 1. Precision and recall curves for the Diff algorithm as a function of K_DIFF, evaluated on the (a) New Orleans and (b) Kashmore floods.

ground conditions and require no human input.

3.1. *Learning Thresholds Automatically*

The Diff, Dart, Fai, and Mndwi algorithms perform reasonably well across a variety of flood conditions, but require a single threshold value to be specified manually. The threshold which achieves the best point on the precision / recall curve varies between domains, and depends on both the appearance of the water and the surrounding land (Fichtelmann and Borg 2012; Klein et al. 2015). For example, Figure 1 shows how differently the same thresholds for the Diff method perform on a flood in Louisiana and a flood in Kashmore.

To eliminate the need for manual human intervention, we introduce an algorithm that computes these thresholds automatically. The key idea is that although the flood conditions vary widely from non-flood conditions, non-flood conditions may still be suitable for estimating a good threshold value, even if that threshold is not perfectly adjusted for the flood conditions.

We search for a MODIS image of the same area from one year before the test date without clouds. If such an image is not available the search is extended to previous years. We choose an image from the same time of year in order to obtain training data with similar seasonal land cover.

Histograms of the water and land pixel values are generated from the historical image, with the 250m MOD44W permanent water mask (Carroll et al. 2009) used as ground truth. Let W_T and L_T be the sets of water and land pixels as thresholded by a threshold T, and W and L be the true sets of water and land pixels according to the permanent water mask. We choose the threshold T that maximizes

$$\frac{|\mathrm{L}_T \cap \mathrm{L}|}{|\mathrm{L}|} \frac{|\mathrm{W}_T \cap \mathrm{W}|}{|\mathrm{W}|} \tag{7}$$

where |A| is the size of set A in order to minimize mislabelings for both classes. We then evaluate the original test image with the threshold T.

This automatic approach for learning thresholds is applied to the MODIS thresholding algorithms Diff, Dart, Mndwi, and Fai in order to create the Diff_L, Dart_L, Mndwi_L, and Fai_L algorithms. These approaches require no human input, but since water in the flood image may have a different appearance than the water in the historical image, they do not select as good of a threshold as a human would.

3.2. *Combining Thresholds with Adaboost*

The automatically learned thresholds require no human supervision. However, there is no uniformly "best" thresholding algorithm. Each performs well in certain situations and poorly in others.

To address the shortcomings of the individual thresholds, we propose an algorithm based on Adaboost (Freund and Schapire 1997) that combines multiple threshold-based classifiers into a single more effective classifier, which is robust and works well across a diverse range of flood scenarios. Like the automatically learned thresholding approaches, this algorithm can be trained on on limited historical data, and does not need to be trained post-flood. In fact, the algorithm is effective even if it has not been trained on a particular location.

The algorithm Ada is shown in Algorithm 3. Ada requires a set of indices I_i (which are functions of MODIS bands), thresholds for these indices K_i, and weights for each individual classifier α_i. Each individual classifier may not be very accurate, but when added together and weighted the classifiers achieve a high precision and recall.

Ada returns an image of real numbers, such that positive pixels are flooded and negative pixels are dry. However, we have found that the classification favors false negatives over false positives. We shift along the precision / recall curve in favor of recall by thresholding the output of Ada by -1, such that pixels with a value over -1 are marked as flooded. See Figure 2 for an example of this precision recall curve.

The indices, thresholds and their weights are learned with the procedure LearnAda shown in Algorithm 4. The LearnAda procedure learns from multiple flood images at once. It learns up to M weak classifiers. (We use $M = 50$, terminating

```
function ADA($\mathbf{B}$, $\boldsymbol{I}$, $\boldsymbol{K}$, $\alpha$)
    $R \leftarrow 0$
    for $i = 1..N$ do
        $\mathbf{C} \leftarrow$ image s.t. $\mathbf{C}(i,j) = \begin{cases} 1 & \text{if } \boldsymbol{I}_i(\mathbf{B})(i,j) \leq K_i \\ -1 & \text{otherwise} \end{cases}$
        $R \leftarrow R + \alpha_i C$
    end for
    return $R$
end function
```

Algorithm 3: The ADA algorithm, which produces a flood map from a MODIS image $\mathbf{B}$, and lists of N MODIS indices $\boldsymbol{I}_i$, thresholds $\boldsymbol{K}_i$, and weights α_i. In the returned image, positive pixel values are classified as flooded pixels.

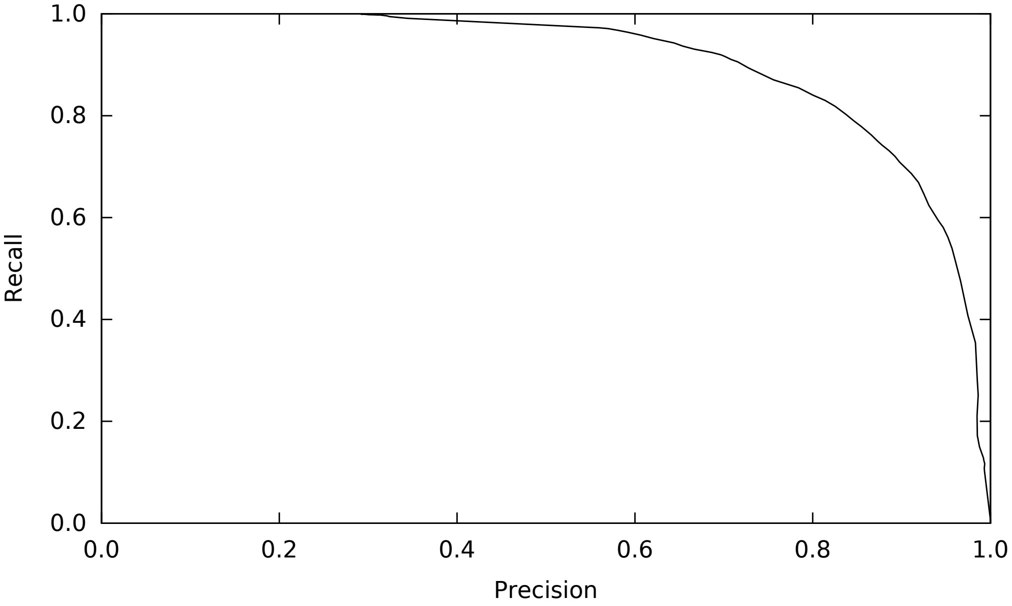


Figure 2. An example precision / recall curve for an image of a flood on the Mississippi in June 2011 (see Section 4.1 for details on the evaluation image).

early if a cycle is reached, but note that each additional classifier is subject to diminishing returns, and overfitting is possible.) Each iteration, a new weak classifier is chosen with the lowest weighted error based on a matrix $\mathbf{W}$ of pixel weightings. We iterate over all possible index functions, and search for thresholds halfway between previously used thresholds. We initiate the list of previous threshold choices with the 20th and 80th percentiles of the flooded pixels in the training data, bounding the searched thresholds to this range.

Once a weak classifier (both index function and threshold) is chosen, a weight α for that classifier is computed. The weights of the image pixels in $\mathbf{W}$ are then updated, such that pixels the selected weak classifier incorrectly classifies are weighted more heavily in the future. Thus, LEARNADA focuses on pixels it previously failed on when learning new classifiers. After M weak classifiers are identified, a list of the classification functions, the selected thresholds, and the classifier weights α is generated. These lists are then used by ADA to map floods.

For MODIS data, we learn from the following index functions:

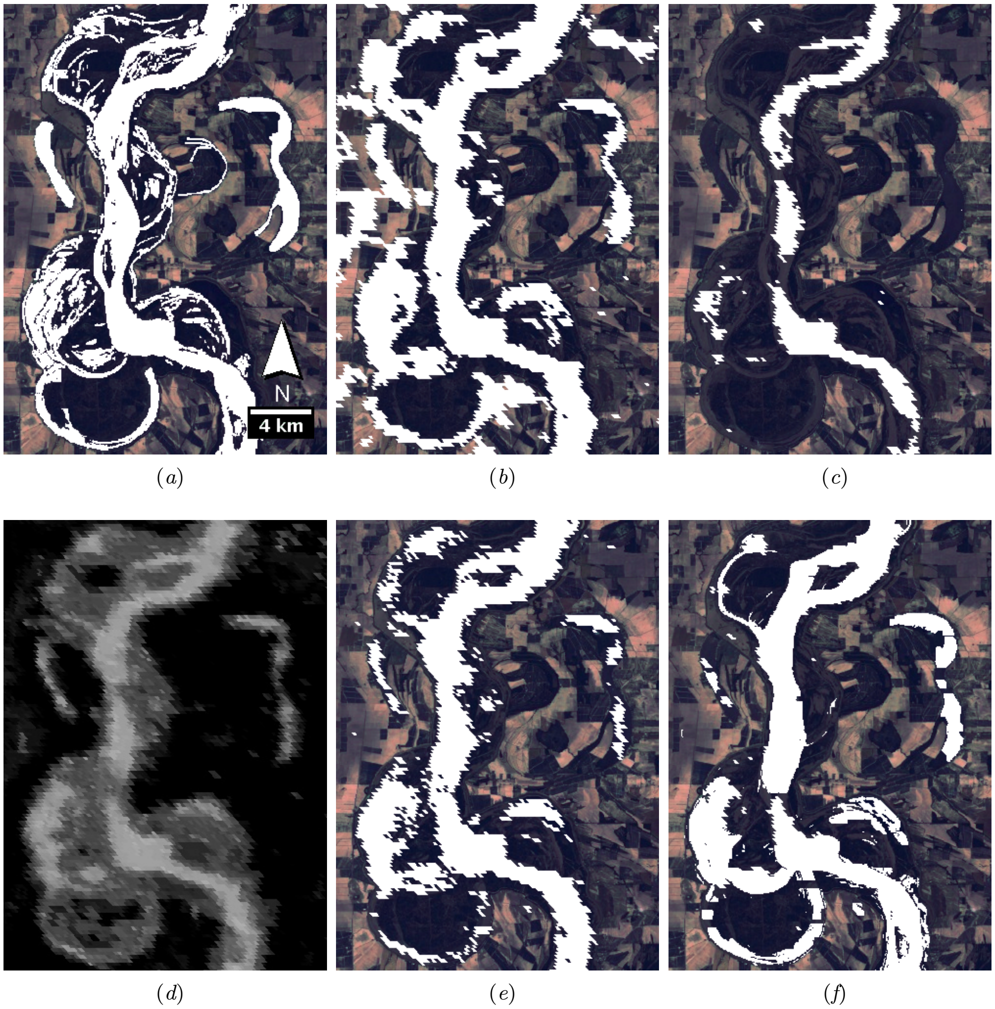
```
function LearnAda($\mathbf{B}$, $\mathbf{T}$)
    $\text{I}_{\text{all}} \leftarrow$ All considered index functions
    for $f \in \text{I}_{\text{all}}$ do
        $a, b \leftarrow$ 20th, 80th percentile in histogram of $f(\mathbf{B})$ s.t. $\mathbf{T}$
        $\boldsymbol{s}_{\text{f}} \leftarrow [a, b, b + (b - a)]$          ▷ List of previously chosen thresholds.
    end for
    $\boldsymbol{I} = [\,]\,, \boldsymbol{K} = [\,]\,, \alpha = [\,]$
    $\mathbf{W} = \frac{1}{|\mathbf{B}|}$          ▷ Matrix of weights for all pixels in $\mathbf{B}$, begin normalized
    for $i = 1..M$ do
                  ▷ Choose classifier and threshold to minimize weighted error
        $(f, j) \leftarrow \arg\min_{f \in \boldsymbol{I}_{\text{all}}, 0 \le j < |\boldsymbol{s}_{\text{f}}|-1} \epsilon = \mathbf{W} \cdot ((f(\mathbf{B}) \le \frac{\boldsymbol{s}_{\text{f},j} + \boldsymbol{s}_{\text{f},j+1}}{2}) \neq \mathbf{T})$
        $K \leftarrow \frac{\boldsymbol{s}_{\text{f},j} + \boldsymbol{s}_{\text{f},j+1}}{2}$
        $\alpha_i \leftarrow \frac{1}{2} \ln \frac{1-\epsilon}{\epsilon}$
        $\forall x, y\ \mathbf{D}(x, y) \leftarrow \begin{cases} 1 \text{ if } f(\mathbf{B})(x, y) \le K \\ -1 \text{ otherwise} \end{cases}$
        $\mathbf{W} \leftarrow e^{-\alpha_i D} \mathbf{W}$          ▷ Update weights
        $\mathbf{W} \leftarrow \mathbf{W} / \sum_{x,y} \mathbf{W}(x, y)$          ▷ Normalize weights
        Append $f, k, \alpha_i$ to $\boldsymbol{I}, \boldsymbol{K}, \alpha$.
    end for
    return $\boldsymbol{I}, \boldsymbol{K}, \alpha$
end function
```

Algorithm 4: This algorithm learns the weak classifiers and weights for use in Ada. $\mathbf{B}$ is a union of potentially multiple input MODIS images, and $\mathbf{T}$ is a ground truth mask indicating which pixels are flooded (0 is dry, 1 is flooded). Operations on images are performed individually on every pixel of the image. Comparison operators on images return 0 for false and 1 for true on a per-pixel basis, and $\mathbf{M} \cdot \mathbf{N}$ is element-wise multiplication.

- B_1
- B_2
- B_2/B_1
- EVI
- LSWI
- NDVI
- NDWI
- LSWI - NDVI
- LSWI - EVI
- Diff
- Dart
- Fai
- Mndwi

Ada uses combinations of all these features for classification. Where a thresholding algorithm is specified, we use the intermediate result in the algorithm that the threshold is applied to.

If further algorithms are developed which output a continuous range of values, Ada can also incorporate them as training features. Ada's inputs do not even need to be from MODIS. In fact, we have successfully incorporated SAR and Landsat data into Ada, by simply adding the additional information as index functions. Then Ada automatically learns which satellites and index functions to incorporate into the classifier. Figure 3 shows an example of Ada and the results of thresholding two of its inputs.



(*a*) (*b*) (*c*)

(*d*) (*e*) (*f*)

Figure 3. The results of several different flood mapping algorithms applied to a flood of the Mississippi. Ada combines multiple approaches and is more robust than individual algorithms. (*a*) Landsat overlaid with ground truth annotation. (*b*) Dart. (*c*) Theme. (*d*) Ada unthresholded output. (*e*) Ada. (*f*) AdaDem. The horizontal artifacts are caused by artifacts in the DEM. All images are from Landsat 5, path 27 row 37, taken on 10 Oct. 2011 with a top left corner at 33.16°N, 91.24°W.

3.3. *Fine-tuning Adaboost with the DEM*

The output of Ada is at MODIS' 250m resolution. However, as with DNNS, we can perform mapping at a higher resolution by incorporating the DEM. In fact, we can apply the exact same ApplyDem algorithm as Ddem uses. We merely shift the output of Ada, adding one to the output and clamping between 0 and 1, and pass that image as the water fraction to ApplyDem. This means that pixels with a value less than -1 (our looser threshold used to adjust the results in favor of recall) in the image from Ada will be considered land, and pixels with a value greater than 0 (the original threshold) are considered water. It is unclear if the assumption behind Dnns, that a pixel with water fraction 0.7 has 70% of its surface area covered with water, holds for this case. Figure 3 shows an example of Ada and AdaDem.

4. Experimental Evaluation and Discussion

We evaluate both Ada and the other MODIS algorithms on six floods. We also evaluate the MODIS flood mapping algorithms on 756 lakes to gain additional confidence and understanding of their capabilities. Our aim is to establish an empirical foundation with which to evaluate flood mapping algorithms, so we can say with certainty whether algorithms are effective across a range of locations and climates. For convenience, a summary of all our experiments is given in Table 2.

Table 2. Summary of experiments. Section and table numbers refer to this paper.

Section	Table	Name	Inputs	Test sites	Ada training data
4.1	4	MODIS floods	MODIS only	6 floods	Permanent water mask on 5 images
4.2	5	MODIS lakes	MODIS only	756 lakes	PWM on 4 images and 12 lakes

Towards this end, all the code for both the algorithms and evaluation is released open source under the Apache 2.0 license (available at https://github.com/nasa/CrisisMappingToolkit). We encourage other researchers to make use of this software to evaluate their own flood mapping algorithms.

4.1. *Selected MODIS Floods Evaluation*

We selected a number of MODIS flood images to evaluate the selected algorithms on.

4.1.1. Evaluation Regions

In order to evaluate the MODIS flood detection algorithms we assembled eight data sets. Each data set was hand picked from the vast MODIS archive available in Earth Engine to meet the following criteria:

(1) Show a significant **flood** event, visible at MODIS' resolution.
(2) Contain minimal or **no clouds**.
(3) Have a low-cloud **Landsat image** available on approximately the same date, for the purposes of establishing ground truth. Since Landsat's repeat cycle is 16-days, a Landsat image is not always available for each flood with a quality MODIS image.

Due to the limited number of Landsat images and the fact that flooding is commonly accompanied by clouds, there are actually few major floods that meet all of these conditions.

The following regions were used to evaluate the MODIS flood mapping algorithms:

(1) **Sava River, Bosnia and Croatia**: The Sava river flooded in May 2014, inundating nearby fields and small villages.
(2) **Kashmore, Pakistan**: The Indus river flooded in August 2010. This is a wide flood across an extended floodplain. The river water appears to be filled with dirt and on some MODIS channels gives a similar response to the nearby desert.
(3) **Mississippi River, Arkansas / Louisiana / Mississippi Border, United States**: The Mississippi river flooded at the Mississippi / Arkansas / Louisiana border in May 2011. This region has multiple isolated bodies

of water and substantial vegetation, making it one of the most challenging regions we address. Due to the vegetative cover this region has considerable uncertainty in the human generated ground truth map.

(4) **Mississippi River, Arkansas / Louisiana / Mississippi Border, United States**: This covers the same region as the previous flood, but a month later when the water level has risen even further. The appearance of the nearby fields has changed considerably as the crops have grown.

(5) **Katrina, New Orleans, United States**: The city of New Orleans was flooded after Hurricane Katrina. This is a densely populated urban area. The flood is difficult to see from Landsat imagery, so maps generated from on-the-ground observations were used to construct the ground truth map.

(6) **Assiniboine River, Manitoba, Canada**: Significant precipitation caused the Assiniboine River to flood in Manitoba. The test area is near the town of St. Lazare. This domain is especially challenging since the river is quite narrow compared to the MODIS resolution, and nearby fields appear as flooded.

(7) **Parana River, Argentina**: The Parana river flooded due to heavy rainfall. The flood region is located near the border with Paraguay, and includes the town of Corrientes.

(8) **Shire River, Malawi**: Malawi suffered a devastating flood in January 2015. The Shire River dramatically exceeded its usual boundaries but the flood water is is broken up by small outcroppings of land. High resolution Skybox RGB imagery aided in generating the ground truth for this region.

Table 3. Test domain details.

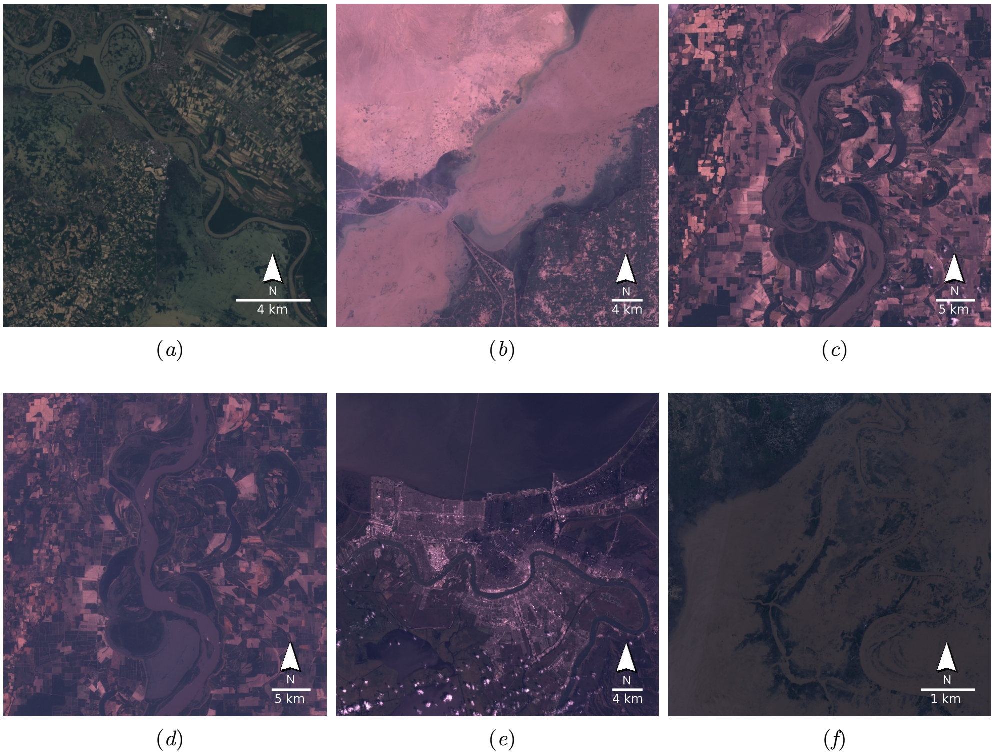
		MODIS	Ground Truth		Coordinates of Bounding box	
ID	Name	Date	Date	Satellite	Bottom left	Top right
1	Sava	21 May 2014	22 May 2014	Landsat 8	44.919°N, 18.507°E	45.101°N, 18.809°E
2	Kashmore	13 Aug. 2010	12 Aug. 2010	Landsat 5	28.250°N, 69.500°E	28.650°N, 70.100°E
3	Mississippi	8 May 2011	10 May 2011	Landsat 5	32.880°N, 91.230°W	33.166°N, 91.020°W
4	Mississippi	12 June 2011	11 June 2011	Landsat 5	32.880°N, 91.230°W	33.166°N, 91.020°W
5	Katrina	7 Sep. 2005	7 Sep. 2005	Landsat 5	29.900°N, 90.300°W	30.070°N, 89.760°W
6	Assiniboine	1 June 2011	1 June 2011	Landsat 5	50.200°N, 101.690°W	50.460°N, 101.330°W
7	Parana	19 July 2014	19 July 2014	Landsat 8	27.750°S, 58.950°W	27.500°S, 58.750°W
8	Malawi	21 Jan. 2015	21 Jan. 2015	Skybox	17.090°S, 35.180°E	16.870°S, 35.320°E

The exact bounding boxes and the sources of the MODIS and ground truth images are listed in Table 3. Partial images of the test areas are shown in Figure 4. Our aim was to select a wide range of flood conditions, representative of the diversity of floods worldwide.

4.1.2. Methodology

For each domain, we evaluate each algorithm numerically against a "ground truth" image by computing the precision P and recall R for the flooded regions. Let T be the set of pixels flooded in the ground truth image, and G be the set of pixels the algorithm guesses are flooded. Then

$$P = \frac{|\mathrm{T} \cap \mathrm{G}|}{|\mathrm{G}|}, R = \frac{|\mathrm{T} \cap \mathrm{G}|}{|\mathrm{T}|}. \tag{8}$$



(*a*) (*b*) (*c*)

(*d*) (*e*) (*f*)

Figure 4. Partial images of the tested flood regions (see Table 3 for complete details). (*a*) Region 1, Sava. (*b*) Region 2, Kashmore. (*c*) Region 3, Mississippi. (*d*) Region 4, Mississippi. (*e*) Region 5, New Orleans. (*f*) Region 8, Malawi.

Precision gives a measure of false positives, while recall gives a measure of false negatives.

"Ground truth" images for evaluation are generated manually through careful inspection of the Landsat images. We currently have more confidence in manual annotation than in any automatic Landsat flood mapping algorithm. However, there are still errors and uncertainty in the manually annotated ground truth images. Areas with vegetation are especially problematic, as the surface underneath the tree canopy is not visible from the Landsat imagery. In these areas, the annotator applies their best guess based on the surrounding context and the DEM, if relevant. Urban areas are also challenging: however, for areas of particular interest to humans, historical records are often available which the annotator can use to help generate the ground truth image.

We evaluate a total of 21 flood mapping algorithms. Of these, 14 were introduced by previous researchers and described in Section 2. We list all implementation details which were unspecified or differ from the original algorithm.

Evi, Xiao, and Theme require no training or thresholds.

Diff, Dart, Fai, and Mndwi require a single threshold to be specified. We select the threshold which gives the largest area under the precision recall curve, maximizing the product of precision and recall. In practice, when the optimal threshold is unknown, these algorithms will perform worse than our reported results. For Dart, we use the parameters $C = 500$, $D = 2500$, which we determined to be effective through experimentation.

Cart, Svm, Rf, Dnns, and Ddem require a training image with ground truth to learn from. For each test image, we select a nearby smaller area on the same Landsat image to use as training data, separate from the test data. We attempt to select a training region of similar land and water composition to the test region. The annotator also establishes ground truth for this region. Training regions are chosen to closely match the composition of the test region as much as possible. These conditions are highly favorable towards algorithms which require human input.

For Cart, Svm, and Rf, built-in Earth Engine classifiers are used. The original Dnns involved a random forest that separated between land, water and "mixed" pixels; however, labeling these three classes is challenging. Instead, we train a probabilistic random forest classifier with only water and land classes, and segment the pixels into three categories based on the output probabilities. Our Dnns implementation additionaly differs from the original in that we use $R_{\mathrm{D}} = 40$ pixels rather than $R_{\mathrm{D}} = 100$ pixels. This is so that the algorithm can finish more quickly. Even with the parallelism of Google Earth Engine, Dnns takes substantially longer than all the other algorithms, which complete in a matter of seconds. Finally, for Ddem, due to limitations of Earth Engine, we use a linear approximation of the histogram percentage computation as shown in Algorithm 2. Additionally, we smooth over all water bodies rather than only connected water bodies, and treat all water with a uniform smoothing radius rather than treating water bodies differently based on their size. As the smoothing radius is small enough that different water bodies rarely overlap, we do not expect this to be a large change.

The remaining algorithms are original to this work. Diff$_{\mathrm{L}}$, Dart$_{\mathrm{L}}$,Fai$_{\mathrm{L}}$, Mndwi$_{\mathrm{L}}$, Ada, and AdaDem were presented in Section 3. To demonstrate the importance of the features selected for Ada, we also test Ada$_{\mathrm{RAW}}$, which is identical to Ada except the features used are the raw MODIS values for the first six bands. Furthermore, we test the algorithms Cart$_{\mathrm{Ada}}$ and Rf$_{\mathrm{Ada}}$ which apply their respective machine learning algorithms to the Ada features, showing that it is not only the feature selection, but the algorithm itself which has an impact on the results.

Ada was trained on four non-flood images: ones of the Mississippi and New Orleans test areas, one year before the floods, one of the San Francisco Bay Area in non-flood conditions, and one nearby the Sava test site in Serbia in non-flood conditions. The same trained classifier was used to evaluate all of the floods. Kashmore was not used because the permanent water mask was displaced, and Malawi was not used for training because the flooded rivers did not show up at all on the permanent water mask. The site in Serbia was used instead of the Sava test site because the Sava river site was only a single pixel wide in the permanent mask.

4.1.3. Evaluation Results

The results from the evaluation are shown in Table 4. The first subdivision of algorithms requires no human input, the second subdivision requires a threshold value to be specified, and the third subdivision requires a classified image region to train on. Note that for the Assiniboine and Malawi domains, the learned threshold algorithms fail because few pixels from the flooded rivers appear on the permanent water mask.

Among the algorithms that do not require human input, Ada or AdaDem has the best performance for five of eight floods. For the remaining three, they are still close to the best performance. Note that every other algorithm which does not require human input performs extremely poorly (precision or recall below 0.50, or not applicable) on at least one flood. The machine learning approaches also all failed by this criteria on at least one flood. This suggests that our Adaboost approach is more robust than other state of the art algorithms against different

Table 4. Results from the algorithm evaluation in terms of precision, P, and recall, R, subdivided by amount of human input required. For each flood and amount of human input, the algorithm with the highest product of precision and recall is bolded.

	Sava		Kashmore		Mississippi May		Mississippi June		Katrina		Assiniboine		Parana		Malawi	
Algorithm	P	R	P	R	P	R	P	R	P	R	P	R	P	R	P	R
Evi	0.98	0.61	**0.97**	**0.95**	0.99	0.28	0.97	0.46	0.99	0.29	0.86	0.03	0.93	0.35	0.46	0.69
Xiao	0.98	0.79	**0.95**	**0.97**	0.96	0.36	0.93	0.49	0.99	0.31	0.45	0.00	0.91	0.30	0.44	0.84
Diff$_{L}$	0.94	0.83	0.76	0.96	0.59	0.85	0.79	0.84	0.95	0.74	—	—	0.91	0.52	—	—
Dart$_{L}$	**0.85**	**0.92**	0.88	0.97	0.63	0.85	0.80	0.83	0.94	0.73	—	—	0.93	0.49	—	—
Fai$_{L}$	0.90	0.78	0.46	0.50	0.54	0.84	0.72	0.83	0.95	0.68	—	—	0.90	0.57	—	—
Mndwi$_{L}$	0.85	0.84	0.84	0.99	0.48	0.95	0.66	0.90	0.87	0.60	—	—	0.91	0.13	—	—
Theme	0.91	0.64	**0.97**	**0.95**	0.98	0.33	0.97	0.47	0.99	0.29	0.85	0.04	0.93	0.38	0.62	0.65
Ada$_{raw}$	0.73	0.98	0.58	0.99	0.72	0.80	0.78	0.85	**0.85**	**0.84**	0.34	0.63	0.69	0.88	0.46	0.90
Ada	0.77	0.96	0.90	0.97	0.79	0.72	**0.83**	**0.82**	0.94	0.73	**0.63**	**0.55**	0.75	0.74	**0.73**	**0.69**
AdaDem	0.76	0.92	0.88	0.89	**0.81**	**0.71**	0.79	0.84	0.77	0.65	0.46	0.60	**0.86**	**0.77**	0.70	0.67
Diff	0.89	0.89	0.95	0.95	0.79	0.75	0.81	0.81	0.85	0.90	0.89	0.34	0.81	0.77	**0.66**	**0.89**
Dart	**0.90**	**0.89**	0.92	0.97	**0.78**	**0.77**	**0.80**	**0.83**	**0.86**	**0.91**	0.72	0.44	**0.78**	**0.80**	0.76	0.50
Fai	0.85	0.84	0.58	0.94	0.67	0.67	0.74	0.76	0.87	0.87	0.62	0.29	**0.80**	**0.78**	0.66	0.86
Mndwi	0.85	0.84	**0.95**	**0.98**	0.70	0.70	0.77	0.77	0.76	0.97	**0.79**	**0.57**	0.69	0.54	0.49	0.87
Cart	0.95	0.73	0.92	0.98	0.95	0.48	0.95	0.52	0.79	0.95	0.76	0.54	0.49	0.67	0.39	0.91
Svm	0.98	0.45	0.62	0.95	0.73	0.13	0.99	0.42	0.85	0.79	0.70	0.21	0.60	0.73	0.35	0.98
Rf	0.93	0.63	**0.96**	**0.96**	0.75	0.41	0.89	0.53	0.78	0.94	0.55	0.52	0.31	0.71	**0.96**	**0.83**
Cart$_{Ada}$	0.30	1.00	0.94	0.01	0.85	0.48	0.78	0.26	0.96	0.50	**0.81**	**0.55**	0.15	0.66	0.78	0.92
Rf$_{Ada}$	0.89	0.63	0.93	0.53	0.70	0.52	0.66	0.31	0.95	0.50	0.68	0.60	0.16	0.67	0.79	0.28
Dnns	**0.91**	**0.90**	0.92	0.97	**0.84**	**0.60**	**0.88**	**0.74**	**0.80**	**0.95**	0.86	0.16	0.72	0.64	0.38	0.97
Ddem	0.74	0.90	0.79	0.95	0.98	0.39	0.94	0.61	—	—	0.85	0.16	**0.71**	**0.74**	0.36	1.0

flood conditions while not requiring any human input. The results of some of the manually specified thresholds, such as Diff and Fai, are nearly on par with Ada, but require human help. Furthremore, in practice humans are unlikely to pick the optimal threshold and performance will degrade.

Notably, Ada also succeeded on images it wasn't trained for specifically, particularly on the Kashmore, Parana, and Malawi data sets. Ada also performed comparably well on the Assiniboine dataset without training, which was especially difficult for all of the algorithms. Cart$_{\text{Ada}}$ and Rf$_{\text{Ada}}$ performed very poorly, indicating that the Adaboost algorithm itself provides benefits, not only the choice of training data. Furthermore, Ada$_{\text{RAW}}$ performed nearly as well as Ada on the datasets it was trained for, but performed poorly on the others. This shows that the choice of training features is important, as features more closely correlated with water lead to a more robust classifier. Although we do not include the results in the chart, we also trained Ada individually on each unflooded image, and it performed only slightly worse than Ada trained on the four datasets, still much better than Cart$_{\text{Ada}}$ and Rf$_{\text{Ada}}$. This confirms that the Adaboost algorithm itself provides an improvement, not only the additional data used for training.

Of the algorithms with human specified thresholds, Dart performed the best. For the algorithms with annotated training data, Dnns, which is based on Rf, outperformed the rest. These threshold algorithms are very competitive with the third set of algorithms and may represent a much better payoff relative to training effort invested.

Note that, despite its intuitive usefulness, adding the DEM to the Ada and Dnns algorithms appears to provide little quantitative benefit. However, the maps are much more aesthetically pleasing with the DEM (see Figure 3). It is unknown if different algorithms could achieve better results with the DEM, or if the DEM simply offers little useful information over MODIS for improved flood mapping.

4.2. *Extensive MODIS Standing Water Evaluation*

The previous experiments demonstrated the suitability of each algorithms for detecting water in flood scenarios. However, due to the rarity of flood events for which a cloudless Landsat image is available, as well as the time-consuming nature of manually annotating floods for evaluation, we are only able to evaluate eight floods. We would ideally like to evaluate the algorithms on hundreds of floods.

Thus, to further evaluate the performance of the algorithms, we also have performed tests on detecting standing water in lakes. Although lakes differ from floods, by evaluating on lakes we aim to show that Ada succesfully detects water in many diverse regions, not only in the limited flood conditions we were able to evaluate with entirely cloud-free MODIS and Landsat images. 756 lake polygons are selected from the Global Lakes and Wetlands Database (Lehner and Döll 2004), restricted to lakes with areas less than 5000 km^2 and latitude between 55° S and 55° N. All of the MODIS algorithms were then run on expanded bounding boxes surrounding these polygons. The results were then evaluated against the permanent water mask. While water levels change over time, especially during floods, this permanent water mask is a good enough approximation of actual water coverage in typical scenarios for algorithm comparison. Images which contain more than 5% clouds are ignored, judged according to mask bits included with the MODIS images. The algorithms are evaluated on each lake on five dates in the summer of 2014: May 1, June 1, July 1, August 1, and September 1.

For the learning algorithms which require training data, the same region from one year earlier is used, and trained on the same permanent water mask. This gives the learned algorithms a heavy advantage, as the training data is very similar to the test data, unlike in the case of a flood. Hence, we fully expect that the learned algorithms will outperform Ada, which is trained on the same data as for the floods, plus an additional twelve randomly selected lakes.

Table 5. Results from the algorithm evaluation on 756 lakes in terms of precision, P, and recall, R. The top subdivision contains algorithms that were not retrained for each lake. The second and third subdivisions were automatically trained for each lake with the second subdivision all using the histogram division training method from Section 3.1. Note that for the second and third subdivisions, the training data is highly similar to the testing data, so these algorithms have a heavy advantage. The algorithm with the highest product of precision and recall for each percentile and subdivision are bolded.

	9th Percentile		25th Percentile		Median		75th Percentile		91st Percentile	
Algorithm	P	R	P	R	P	R	P	R	P	R
Evi	0.33	0.11	0.71	0.28	0.90	0.48	0.98	0.68	1.00	0.86
Xiao	0.36	0.15	0.71	0.32	0.90	0.52	0.97	0.73	0.99	0.90
Theme	0.33	0.12	0.71	0.30	0.90	0.49	0.98	0.70	1.00	0.87
Ada	0.44	0.65	**0.66**	**0.83**	**0.81**	**0.93**	**0.90**	**0.98**	**0.95**	**1.00**
AdaDem	**0.37**	**0.81**	0.59	0.92	0.73	0.98	0.84	1.00	0.92	1.00
Diff$_{\mathrm{L}}$	**0.37**	**0.67**	**0.62**	**0.86**	**0.82**	**0.93**	**0.93**	**0.97**	0.98	0.99
Dart$_{\mathrm{L}}$	0.36	0.56	0.59	0.85	0.81	0.93	**0.93**	**0.97**	**0.98**	**1.00**
Fai$_{\mathrm{L}}$	0.28	0.58	0.52	0.81	0.74	0.90	0.89	0.95	0.96	0.98
Mndwi$_{\mathrm{L}}$	0.27	0.58	0.53	0.79	0.77	0.90	0.90	0.95	0.96	0.98
Cart	0.63	0.54	0.82	0.77	**0.92**	**0.89**	**0.96**	**0.96**	0.99	0.98
Svm	0.65	0.01	0.85	0.34	0.97	0.69	1.00	0.87	1.00	0.96
Rf	0.47	0.50	0.66	0.74	0.82	0.86	0.92	0.94	0.96	0.97
Dnns	**0.72**	**0.52**	**0.88**	**0.75**	0.94	0.86	0.97	0.95	**0.99**	**0.99**
Ddem	0.55	0.75	0.72	0.89	0.84	0.96	0.91	0.99	0.96	1.00

The results are shown in Table 5 at the 9th, 25th, 50th, 75th, and 91st percentiles of the precision and recall across all the lakes on multiple dates. This presentation allows a glimpse of the distribution of precision and recall. However, note that low precision does not correlate with low recall (in fact, it is the opposite) and in this presentation the relationship between the precision and recall is not shown. For some lakes and algorithms, Earth Engine returns an error (likely because the computation took too long), and we exclude any such data points from the results. Evaluating all 756 lakes in Earth Engine took several days, although it could easily be sped up with more threads since the problem is embarrassingly parallel.

The results suggest that Ada performs decently well on 75% of the lakes. It greatly outperforms the Evi, Xiao, and Theme algorithms. Shockingly, Ada does approximately as well or slightly better than the approaches Diff$_{\mathrm{L}}$, Dart$_{\mathrm{L}}$, Fai$_{\mathrm{L}}$, and Mndwi$_{\mathrm{L}}$, despite the fact that these algorithms have the enormous advantage of having a threshold which was trained on very similar data. Only Cart and Dnns outperform Ada by a significant margin (with their advantage of training on data so similar to the test data), and Ada is not too far behind.

From the lakes study, we can conclude that Ada is fairly robust, with median values of 0.81 for precision and 0.93 for recall. Note that for these results we assume that the permanent water mask is correct, which may help account for some of the failure cases for all the algorithms.

5. Conclusion

We have quantitatively evaluated a number of state-of-the-art flood mapping algorithms across a variety of flood conditions. Our software has been released open source so that anyone can quickly and easily evaluate their own novel flood mapping algorithms.

Our new algorithm, based on Adaboost, combines multiple thresholding algorithms into a single classifier. It has proven effective at mapping floods from MODIS data without any human input, potentially enabling the development of improved fully automatic rapid response flood mapping tools. Adaboost is also effective across a variety of flood conditions.

To transition Adaboost from a prototype to a production level, fully automatic flood mapping system, further effort is needed. First, the location of floods must be known so that data can be collected and maps can be made. Flood locations can be determined from offical flood warnings or crowdsourced with georeferenced tweets. Once a flood location is known, additional data can be collected automatically from other sources such as SAR and Landsat to be fused with the MODIS data by incorporating these sources as additional weak classifiers in Adaboost. Once the images are acquired, clouds must be removed from the MODIS images so that they are not detected as floods, with an algorithm such as Fmask (Zhu and Woodcock 2012). In the case of cloud occlusions, the flood mapping algorithm could rely solely on the other data sources. Alternatively, Adaboost could be trained on images of clouds, although this would necessitate additional training data.

Although many challenges remain for a fully automatic system, Adaboost demonstrates the feasibility of an effective automatic flood mapping system.

Acknowledgements

We are grateful to our partners at Google, especially Josh Livni, Pete Giencke, and Noel Gorelick, for their support in this work. We would also like to thank Cristina Milesi and Larry Barone from NASA Ames, John Stock and Marie Peppler from USGS, and Pat Cutler from NGA for their helpful advice and encouragement.

References

Alfieri, L., P. Burek, E. Dutra, B. Krzeminski, D. Muraro, J. Thielen, and F. Pappenberger. 2013. "GloFAS–Global Ensemble Streamflow Forecasting and Flood Early Warning." *Hydrology and Earth System Sciences* 17 (3): 1161–1175.

Barnes, William L, Thomas S Pagano, and Vincent V Salomonson. 1998. "Prelaunch characteristics of the moderate resolution imaging spectroradiometer (MODIS) on EOS-AM1." *Geoscience and Remote Sensing, IEEE Transactions on* 36 (4): 1088–1100.

Boschetti, M., F. Nutini, G. Manfron, P.A. Brivio, and A. Nelson. 2014. "Comparative Analysis of Normalised Difference Spectral Indices Derived from MODIS for Detecting Surface Water in Flooded Rice Cropping Systems." *PLOS ONE* 9 (2): e88741.

Brakenridge, G.R. 2012. "Technical Description, DFO-GSFC Surface Water Mapping Algorithm." http://floodobservatory.colorado.edu/Tech.html.

Brakenridge, R., and E. Anderson. 2006. "MODIS-Based Flood Detection, Mapping and Measurement: The Potential for Operational Hydrological Applications." In *Transboundary Floods: Reducing Risks Through Flood Management,* 1–12. Springer.

Breiman, L. 2001. "Random Forests." *Machine Learning* 45 (1): 5–32.

Carroll, M.L., J.R. Townshend, C.M. DiMiceli, P. Noojipady, and R.A. Sohlberg. 2009. "A

New Global Raster Water Mask at 250m Resolution." *International Journal of Digital Earth* 2 (4): 291–308.

Farr, T.G., P.A. Rosen, E. Caro, R. Crippen, R. Duren, S. Hensley, M. Kobrick, et al. 2007. "The Shuttle Radar Topography Mission." *Reviews of Geophysics* 45 (2).

Feng, L., C. Hu, X. Chen, X. Cai, L. Tian, and W. Gan. 2012. "Assessment of Inundation Changes of Poyang Lake Using MODIS Observations Between 2000 and 2010." *Remote Sensing of Environment* 121: 80–92.

Fichtelmann, Bernd, and Erik Borg. 2012. "A new self-learning algorithm for dynamic classification of water bodies." In *Computational Science and Its Applications–ICCSA 2012,* 457–470. Springer.

Freund, Y., and R.E. Schapire. 1997. "A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting." *Journal of Computer and System Sciences* 55 (1): 119–139.

Frey, R.A., S.A. Ackerman, Y. Liu, K.I. Strabala, H. Zhang, J.R. Key, and X. Wang. 2008. "Cloud Detection with MODIS. Part I: Improvements in the MODIS Cloud Mask for Collection 5." *Journal of Atmospheric and Oceanic Technology* 25 (7): 1057–1072.

Gesch, D., M. Oimoen, S. Greenlee, C. Nelson, M. Steuck, and D. Tyler. 2002. "The National Elevation Dataset." *Photogrammetric Engineering and Remote Sensing* 68 (1): 5–32.

Islam, A.S., S.K. Bala, and A. Haque. 2009. "Flood Inundation Map of Bangladesh Using MODIS Surface Reflectance Data." In *International Conference on Water and Flood Management (ICWFM),* Vol. 2739–748.

Klein, Igor, Andreas Dietz, Ursula Gessner, Stefan Dech, and Claudia Kuenzer. 2015. "Results of the Global WaterPack: a novel product to assess inland water body dynamics on a daily basis." *Remote Sensing Letters* 6 (1): 78–87.

Kugler, Z., and T. De Groeve. 2007. "The Global Flood Detection System." *Office for Official Publications of the European Communities: Ispra, Italy* .

Kugler, Z., T. De Groeve, and B. Thierry. 2007. "Towards a Near-Real Time Global Flood Detection System." *European Commission Joint Research Centre* .

Lehner, B., and P. Döll. 2004. "Development and Validation of a Global Database of Lakes, Reservoirs and Wetlands." *Journal of Hydrology* 296 (1): 1–22.

Li, S., D. Sun, M. Goldberg, and A. Stefanidis. 2013a. "Derivation of 30-m-resolution Water Maps from TERRA/MODIS and SRTM." *Remote Sensing of Environment* 134: 417–430.

Li, S., D. Sun, Y. Yu, I. Csiszar, A. Stefanidis, and M.D. Goldberg. 2013b. "A New Short-Wave Infrared (SWIR) Method for Quantitative Water Fraction Derivation and Evaluation with EOS/MODIS and Landsat/TM data." *IEEE Transactions on Geoscience and Remote Sensing* 51 (3): 1852–1862.

Martinis, S. 2010. "Automatic Near Real-Time Flood Detection in High Resolution X-Band Synthetic Aperture Radar Satellite Data Using Context-Based Classification on Irregular Graphs." Ph.D. thesis. LMU.

Martinis, S., A. Twele, C. Strobl, J. Kersten, and E. Stein. 2013. "A Multi-Scale Flood Monitoring System Based on Fully Automatic MODIS and TerraSAR-X Processing Chains." *Remote Sensing* 5: 5598–5619.

Martinis, S., A. Twele, and S. Voigt. 2009. "Towards Operational Near Real-Time Flood Detection Using a Split-Based Automatic Thresholding Procedure on High Resolution TerraSAR-X Data." *Natural Hazards and Earth System Science* 9 (2): 303–314.

Matgen, P., R. Hostache, G. Schumann, L. Pfister, L. Hoffmann, and H.H.G. Savenije. 2011. "Towards an Automated SAR-Based Flood Monitoring System: Lessons Learned from Two Case Studies." *Physics and Chemistry of the Earth, Parts A/B/C* 36 (7): 241–252.

Olshen, L., and C.J. Stone. 1984. "Classification and Regression Trees." *Wadsworth International Group* .

Pekel, J-F, C Vancutsem, Lucy Bastin, M Clerici, Eric Vanbogaert, E Bartholomé, and Pierre Defourny. 2014. "A near real-time water surface detection method based on HSV transformation of MODIS multi-spectral time series data." *Remote sensing of environ-*

ment 140: 704–716.

Schumann, G., P.D. Bates, M.S. Horritt, P. Matgen, and F. Pappenberger. 2009. "Progress in Integration of Remote Sensing– Derived Flood Extent and Stage Data and Hydraulic Models." *Reviews of Geophysics* 47 (4).

Shalev-Shwartz, S., Y. Singer, N. Srebro, and A. Cotter. 2011. "Pegasos: Primal Estimated Sub-Gradient Solver for SVM." *Mathematical Programming* 127 (1): 3–30.

Sun, D., Y. Yu, and M.D. Goldberg. 2011. "Deriving Water Fraction and Flood Maps from MODIS Images Using a Decision Tree Approach." *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 4 (4): 814–825.

Taubenböck, H., M. Wurm, M. Netzband, H. Zwenzner, A. Roth, A. Rahman, and S. Dech. 2011. "Flood Risks in Urbanized Areas– Multi-Sensoral Approaches Using Remotely Sensed Data for Risk Assessment." *Natural Hazards and Earth System Sciences (NHESS)* 11: 431–444.

United States Geological Survey. 2006. "USGS Fact Sheet 2006-3026: Flood Hazards— A National Threat." .

United States Geological Survey. 2015. "U.S. Geological Survey Flood Inundation Mapping Science." http://water.usgs.gov/osw/flood_inundation.

Wu, H., R.F. Adler, Y. Tian, G.J. Huffman, H. Li, and J. Wang. 2014. "Real-Time Global Flood Estimation Using Satellite-Based Precipitation and a Coupled Land Surface and Routing Model." *Water Resources Research* 50 (3): 2693–2717.

Xiao, X., S. Boles, S. Frolking, C. Li, J.Y. Babu, W. Salas, and B. Moore III. 2006. "Mapping Paddy Rice Agriculture in South and Southeast Asia Using Multi-Temporal MODIS Images." *Remote Sensing of Environment* 100 (1): 95–113.

Zhu, Zhe, and Curtis E Woodcock. 2012. "Object-based cloud and cloud shadow detection in Landsat imagery." *Remote Sensing of Environment* 118: 83–94.