SMU Data Science Review

Volume 2 | Number 1 Article 12

2019

# Repairing Landsat Satellite Imagery Using Deep Machine Learning Techniques

Griffin J. Lane
*SMU*, griffin.is.now@gmail.com

Patricia Goresen
pgoresen@smu.edu

Robert Slater
*Southern Methodist University*, rslater@smu.edu

# Repairing Landsat Satellite Imagery Using Deep Learning and Machine Learning Techniques

Griffin Lane[1], Patricia Goresen[1] , Robert Slater[2]
Master of Science in Data Science, Southern Methodist University,
Dallas, TX 75275 USA
{glane, pgoresen, rslater}@smu.edu

**Abstract.** Satellite Imagery is one of the most widely used sources to analyze geographic features and environments in the world. The data gathered from satellites are used to quantify many vital problems facing our society, such as the impact of natural disasters, shore erosion, rising water levels, and urban growth rates. In this paper, we construct machine learning and deep learning algorithms for repairing anomalies in the Landsat satellite imagery data which arise for various reasons ranging from cloud obstruction to satellite malfunctions. The accuracy of GIS data is crucial to ensuring the models produced from such data are as close to reality as possible. Reducing the inherent bias caused by the obstruction or obfuscation of reflectance values is a simple but effective way to more closely represent the reality of our environment with satellite data. Using clean pixels from previously acquired satellite imagery, we were able to model the bias present in each scene at different times and apply algorithms to fix the inconsistencies. The machine learning model decreased the mean absolute error by an average of 80.1% compared to traditional repair algorithms such as mosaicking.

## 1 Introduction

Due to developments in the past few decades regarding the aerospace industry and information systems, there has been a significant increase in the availability as well as demand for satellite imaging data. In 2008, the United States Geological Survey (USGS) implemented a free-and-open data policy in which they released their Landsat satellite imagery repositories to the public free of charge. Since the policy was enacted, Landsat downloads have increased exponentially. In 2007 there were about 6 million images downloaded with 600 publication citations and in 2017, the number jumped to about 19 million images downloaded with almost 1600 publication citations [1].

The Landsat program is one of the longest running and most widely utilized initiatives for geographic data analysis. Landsat is the most viable source for free, calibrated, and moderate spatial resolution measurements of about 30 meters per pixel of nearly the Earth's entire surface [2]. It contains imagery of Earth's surface over the past 45 years with more than eight million images. The data is binned into wavelength ranges called spectral reflectance bands that record the amplitude of that wavelength reflected off of Earth's surface. We will further discuss how the data is stored in Section 3.1.

These data derived from these satellites are used to drive vital decision-making processes in various levels of commerce and government. The Australian Government uses historical Landsat data to create flood warning maps in order to provide the most accurate warnings possible to its citizens [3]. This data is also used for climate change research, natural disaster relief and impact, optimizing agricultural crop yields, and even cancer research [4]. Researchers rely on

Landsat data to build their models and research. Inaccuracies in the data could greatly affect the models generated and thus affect the results and conclusions derived from such research.

Inaccuracies in satellite imagery data are universal and are inherent features due to various factors. There are two main sources of errors: sensor malfunctions and cloud coverage. Impulse noise, which causes sensors to register bands' frequencies notably higher or lower than the true value, is a widespread problem with Landsat data [5]. There also has been data loss due to satellite equipment failing altogether. For example, Landsat 7 endured a failure in its scanline corrector mechanism. Since then, the ETM+ sensor traces a zig-zag pattern along the satellite's ground track which results in missing data gaps of increasing magnitude. Since then, all data from Landsat 7 has a 22% loss of their pixels and is easily observable in images [6].

By far, the largest contributing source of Landsat data anomalies is cloud coverage. The presence of clouds in satellite imagery is, unfortunately, an inevitable occurrence due to natural weather. As with scanline errors, cloud coverage significantly inhibits the usability and accuracy of the data. Approximately two-thirds of Earth's surface is continually covered by clouds and additionally, about thirty-five percent of pixels have cloud coverage [7-8]. As a result, this creates significant limitations in the satellite's ability to properly scan the Earth's surface. Detecting the presence of clouds in a scene can be somewhat computationally expensive and time-consuming but fortunately, quality assessment metadata for cloud coverage is provided by USGS. The Landsat Collection 1 Quality Assessment applies integer bit values to each pixel. These different bit values represent the condition of that pixel and can be used to determine whether or not a cloud is present in a pixel. We will use this quality band containing USGS' assessment to mask out pixels containing clouds, which we will refer to as dirty pixels. It should be mentioned that there are many algorithms used for cloud detection and repair and many outperform USGS' standard. However, due to the ease of access of use and speed, we will use USGS' algorithm and simply mask out bit values that are not considered clean.

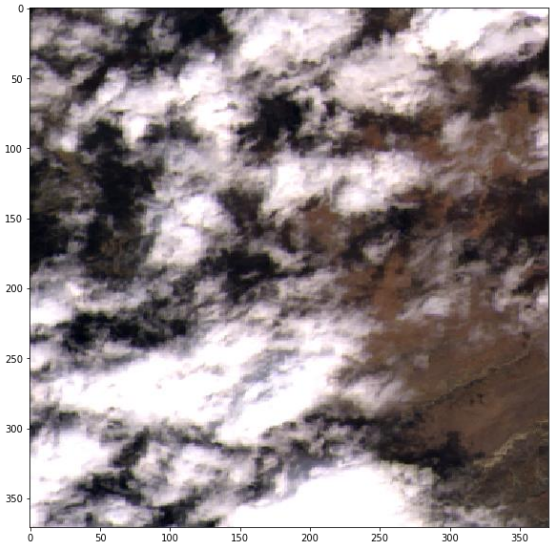


**Figure 1.** Clouds obstructing a clear view to the earth's surface, blocking OLI-TIRS readings.

There are currently numerous methods for repairing Landsat data, see Section 2.1. The most commonly used method uses historical pixel data for a scene to fill in missing data in the scene

at a future target time [9]. Usually, the most recent pixels that are clean are used. This is known as mosaicking. However, there is an easily observable difference in the values where the scenes have been patched. If a band's values are observed, the patch is characterized by either higher or lower values.

In order to combat this problem, we will train a machine learning and a neural network algorithm to model the bias between images for the pixels present. We will then use that model to apply a transformation to the pixels to be used in the mosaic in order to more effectively simulate the missing pixels with the patch. This will create a more consistent and accurate representation of the Earth's surface. Our goal is not only making this process seamless and accurate, but we will also take into consideration the time and cost needed to run our algorithm.

Due to such consideration, the algorithm proposed is not a drop-in replacement for mosaicking but rather a slower but more accurate alternative. The proposed solution would most effectively be utilized as an additional step after downloading and ingesting into a database. This would allow for quick access to the repaired data since all the processing will have been taken care of during the ingestion phase.

# 2 Data Repair Algorithms

## 2.1 Issues with Current Methods

Before proposing another algorithm for repairing gaps in Landsat data, it makes sense to highlight some drawbacks in the current methodologies we are seeking to alleviate. While there are numerous methodologies being utilized to repair Landsat data, many of these methods are not suitable for the creation of distributions over time in particular. Since Landsat 7 has several existing solutions for the reparation of scan line errors, we will focus on Landsat 8 data. Our proposed solution, though likely able to correct scan line errors, does not take advantage of the nearby clean pixels near the gaps of the data. It is likely those pixels could be utilized in a more effective manner. For clouds, however, the gaps are far larger and inconsistently sized and these tools are not suitable.

On average, clouds obstruct an estimated thirty-five percent of all Landsat data globally due to natural weather [7]. Fortunately, USGS provides a quality band which uses a bit value to give the researcher metadata about the pixel. Often, researchers will simply mask out values based on their bit value in that band. Though when clouds are excluded via masking, large portions of a scene are rendered unsuitable for analysis. Thus if pixels are masked out, the researcher does not have the full scene for use in the analysis of the region. To combat this problem, there are a variety of algorithms available for reparation of that data.

Most often, a simple process called mosaicking is used. This method is quick but highly inaccurate as it uses unaltered historical pixel data to patch the gaps in the target scene. When the plotting the bands generated using this method, the patch usually stands out starkly against the true values for that timeslice. This method is better suited for when only one timeslice is required and the analysis being performed is simple land classification, water detection, or any other task that uses relationships between the values but does not rely on the actual values themselves. Even if used in succession for each timeslice it is unsuitable for forming a distribution

since the typical yearly trend would be delayed and the actual peak values may not be represented in the distribution. A notable strength of this method, however, is that it preserves landforms and other structures [10]. For this reason, it will be the basis of our solution.

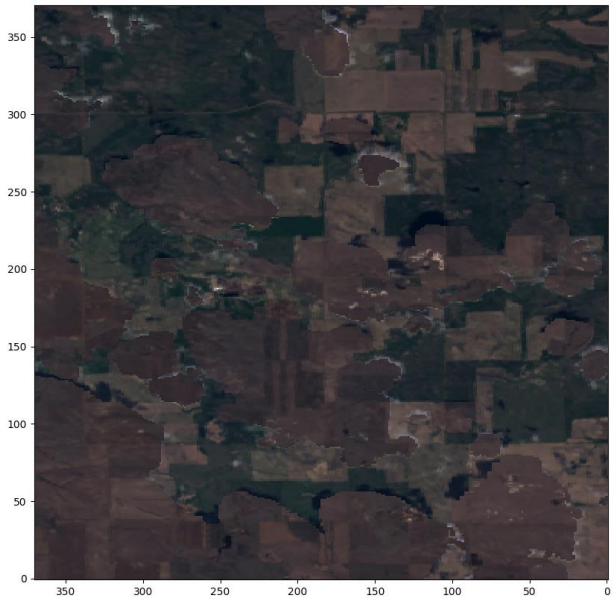


**Figure 2.** A typical GIS mosaic patch created from the most recently available historical timeslice's clean pixels. Note the discoloration throughout the image from the historical slice replacing cloud data.

Another frequently utilized method is to omit the affected pixels. This is common when trying to map the distribution of a derived spectral index, such as the Normalized Difference Vegetation Index (NDVI) over the course of a year as mosaicking would skew the distribution. While the guarantee of clean pixels is convenient, the main drawback of this method is that large portions of scenes can be unusable while the usable parts could be in different parts of the scene. This creates the potential to create a distribution that is not representative of any part of the scene. If each timeslice has clean pixels in a different area, this method of repair is forming a distribution that has observations from different geographic locations and is thus not representative. When gaps are small enough, neighboring pixels can be copied into the missing pixels' locations. However, the larger the gap, the more likely these values are to deviate dramatically from the actual values that would be observed had the instrument not been obscured.

# 3 Data Sources

## 3.1 Landsat Data

The spectral reflectance data we will be using is derived from the United States Geological Survey's Landsat Program, hereafter referred to as USGS Landsat data or simply Landsat data.

The Landsat Program is a cooperative effort by both NASA and the USGS that is designed to make a continually updated temporal record of the Earth's surface as it changes. It provides moderate-resolution, approximately 30 meters per pixel, of multispectral data reflected off of the Earth's surface [2]. The Landsat data consists of spectral characteristics, instruments used, calibration, coverage, and geometric characteristics of all landmasses and near-coastal areas on Earth [2]. These scenes are captured over 650 times a day and are extremely accurate.

The spectral reflectance data is more than just imagery and, in an effort to make that differentiation clearer, acquisitions are commonly referred to as scenes instead of images. Typically, images would only consist of red, blue, green, and possibly an alpha value that would also typically come in a value range of one to two-hundred and fifty-five be it numeric or in a hexadecimal format. The wavelength ranges are illustrated in Table 1. Spectral reflectance is much different and comes in a much higher precision range. It also includes a number of non-visible wavelength bands, like short wave infrared and near-infrared, and could even include radar or thermal bands. Various spectral indices can be derived from this spectral reflectance data and can be used to detect and compare various aspects of the region including, but not limited to, urbanization, the presence of water, and relative vegetation levels. We will be using a combination of these indices in order to get a better overall description of what the environment is like. For instance, an area with a high value of a Normalized Difference Vegetation Index is a great way to get an approximate idea of the relative amount of vegetation.

**Table 1.** Parameters of Landsat Data

| Bands | Wavelength (micrometers) | Resolution (meters) |
|---|---|---|
| Ultra Blue (Coastal/Aerosol) | 0.42 – 0.45 | 30 |
| Blue | 0.45 – 0.51 | 30 |
| Green | 0.52 – 0.59 | 30 |
| Red | 0.64 – 0.67 | 30 |
| Near Infrared (NIR) | 0.85 – 0.88 | 30 |
| Shortwave Infrared (SWIR) 1 | 1.57 – 1.65 | 30 |
| Shortwave Infrared (SWIR) 2 | 2.11 – 2.29 | 30 |
| Panchromatic | 0.50 – 0.68 | 15 |
| Cirrus | 1.36 – 11.19 | 30 |
| Thermal Infrared (TIRS) 1 | 10.60 – 11.19 | 100 |
| Thermal Infrared (TIRS) 2 | 11.50 – 12.51 | 100 |

### 3.2 Landsat Platforms

There are various satellite platforms available under the Landsat Program. The platforms are numbered and range from Landsat 1 to Landsat 8, which is the most recent platform. Each

satellite platform has its own band designations and specialized instruments. The numbers are in order of launch time with the first program being Landsat 1 which commenced in July 1972 and the most recent being Landsat 8 which was developed in February 2013. Since technology has greatly improved since 1972, it makes sense to opt for the use of the latest iteration of the Landsat series of satellites. Since Landsat 8 consists of 11 different bands, it also offers the widest range of possible derived spectral indices.

The instruments on Landsat 8 which are of interest to us are known as the Operational Land Imager (OLI) and Thermal Infrared Sensor (TIRS) which will be labeled as OLI-TIRS in the metadata file associated with the acquisition.

### 3.3 Landsat Tiers

Since 2016, the satellite scene collections have been divided into various tiers. When the satellite data is retrieved, it goes under a quick preprocessing step and becomes immediately available under the first tier, the Near Real Time collection. The next tier is Level1 and consists of data that meet the formal quality criteria for geometric and radiometric geospatial data. The data included in that tier is of verified quality and meets the accuracy requirement of 12m, this accuracy is measured via the root mean squared error (RMSE) after being compared to various ground control points at established locations around the globe. Level 2 is the tier for data that do not fall into the required 12m accuracy geometric and radiometric quality standards. Level 2 generally consists of acquisitions from older satellites as newer satellites have instruments with much higher precision and accuracy. Since Level-1 data is of the highest quality and is suitable for per pixel analysis, we will be using Level-1 collections for our purposes, which will be discussed in Section 3.4.

### 3.4 Landsat Processing

Landsat Level-1 data is processed by the highest available processing level. The highest available processing level is determined by the presence of Digital Elevation Model data (DEM data), the presence of Ground Control Points (GCPs), and potentially the use of Payload Correction Data (PCD) from the satellite itself. We will be using Level-1 TP, or L1TP, data as it is the highest quality of Level-1 data available. L1TP data has been radiometrically calibrated and orthorectified using the GCPs available. DEM data is also used to correct relief displacement due to the variations in elevation and the effects of viewing the surface from the perspective of the aperture. Other levels of processing include L1GT and L1GS, however, a discussion of these levels of processing is beyond the scope of this document, though these levels are used when the information to process at the L1TP level is not present. Since we are only focusing on the highest available quality scenes from Landsat 8 we will have constant confidence intervals for the range of the spatial displacement of each pixel, less than 12m off. Often, the actual spatial displacement is below that 12m maximum.

### 3.5 Landsat Data Source

Conveniently, Amazon Web Services (AWS) hosts all Landsat 8 scenes from the earliest scene to the most recent scenes on a public-facing S3 bucket. It is updated usually within a few hours

of the newest scenes' availability from USGS. To access this S3 bucket data, we will create a query from the command line. We will also be using Amazon's AWS-CLI, a command line interface useful for interacting with AWS. The scenes in the S3 bucket will be indexed into a PostgreSQL database belonging to the Open Data Cube so that the data can be accessed conveniently via the ODC Python API's load function. The structure of the USGS data on the S3 bucket is well documented and straightforward to index.

## 4 Solution Methodology

Bias modeling has been used historically to correct differences in data from different instruments [11-12]. The purpose of bias modeling is to model the inherent differences between the instruments and adjust the readings accordingly in order to create a more uniform and accurate output when matched with the target instrument. For our solution, we tested two different statistical machine learning models. The first model was a standard feed-forward neural network that was intended to pick up on the relationships between the band values and leverage those for predictive purposes. The other model selected was a Random Forest Regression model. Random Forest models are known for being able to handle very different datasets moderately well. This made sense since our training size and band values are likely to fluctuate wildly from region to region.

When an image has missing or unclean data, the data must be omitted or imputed. The current, most widely used imputation algorithm is mosaicking, detailed in Section 2. The mosaicking algorithm fills the gaps in clean data with the most recent chronological timeslice's clean data at the corresponding pixel location. In areas with continual cloud coverage, pixels can be filled with data from several timeslices back. In our data, the maximum number of historical timeslices required was seven. Our two models will model the bias present between timeslices at the corresponding pixel location. The models will then be used to apply a transformation to the standard mosaicking method's resulting pixel patches to achieve a greater approximation of the actual values.

In order to establish the training data and validation data, a series of random pixels will be falsely labeled as dirty pixels and removed from the training process. These viable pixels from the target scene will be cut out and set aside for comparison to the transformed pixels from the patch created from the historical timeslice using simple mosaicking. With that portion set aside, the remaining pixels shared between that historical slice and the target slice are the training set. After the patch has been transformed, if any unclean pixels remain in the target image that was not able to be filled from the historical timeslice, the next most recent timeslice's pixels will be used, and so forth, until all pixels are accounted for.

In order to prevent training on successively imputed data, the most recent timeslice should be the initial target timeslice. Otherwise, the ability to apply this algorithm to all timeslices loaded into memory without imputing values using imputed data is forfeit. For these reasons, it is logical to start from one end or the other chronologically. Theoretically, any timeslice could be used to impute the data since this method does not directly rely on the temporal relationship of this data but, the potential for dramatic environmental change increases directly with the increased distance in time. This means two timeslices with very few shared pixels between, and thus a small amount of training data can be omitted temporarily from the process in favor of using a timeslice with more shared clean pixel locations with the target timeslice.

Another reason for the use of the previous timeslice is because the average span between scene captures from Landsat 8 is sixteen days and the ratios between the values are unlikely to have changed much even if the amount of light reflecting back is greater or less due to the time of day. In this way, we can avoid using an imputed timeslice's data for the imputation of another timeslice.

### 4.1 Neural Network Modeling

The neural network model analyzed in our proposal is a standard feedforward neural network. This model is intended to approximate the function between the input and output bands by learning the mappings between the two timeslices. The input bands will be the eleven spectral reference bands from the clean data in the previous timeslice. The input neurons are then sent into a series of hidden layers where the outputs of all synapses are input into each hidden layer consecutively. In the hidden layers, an activation function is then applied to the weighted sum and passed on to the next hidden layer. This repeats until the model is able to predict the frequencies accurately.

Neural Networks are famous for finding relationships between data and leveraging those relationships for predictive and classification purposes. Unfortunately, any relationships discovered are difficult to interpret so it is considered a black box. Since this research study is not necessarily concerned with interpreting the relationships between the bands and considers this a preprocessing step to such work, this is acceptable. For these reasons, it was deemed a suitable candidate for the repair of the affected data.

### 4.2 Regression Modeling

The Random Forest Algorithm is an incredibly simple yet powerful machine learning algorithm. It tends to perform pretty well on most predictive tasks with minimal hyperparameter tuning. Another major advantage of the random forest algorithm is it allows for a matrix as output. This allows for one model to predict all eleven bands, compared to creating a total of eleven models to predict each band in a single area in other algorithms. For these reasons, it seemed particularly well suited for our solution.

Similar to the neural network model, the random forest regression model is an additive model that makes predictions based on a sequence of models. The random forest algorithm is generally successful at capturing non-linear relationships between the input and target variables.

## 5 Data Ingestion

### 5.1 Xarray Format

Xarray is a python package that is designed to extend the capability of Pandas, a common Python package used for data science and statistical analysis, to be more convenient for use with multidimensional arrays. It is similar to hierarchical indexing in Pandas, also known as multi-indexing, but it allows for naming the dimensions and doesn’t use fixed dimensional arrays as

Pandas' core data types do. These features make xarray an ideal choice for the physical sciences and geospatial data. In this case, the named dimensions will be latitude, longitude, and time. There are alternatives that could provide similar functionality for GIS data, GeoPandas for example, but discussion of these alternatives is beyond the scope of this document.

### 5.2 Loading Landsat 8 GeoTIFFs via Open Data Cube

Landsat data from USGS is stored in GeoTIFF format. GeoTIFFs are a standard format for GIS data and useful because they contain not only the values for each pixel in the scene but also metadata about the scene, instruments used, satellite platform, data format, reprojection, spatial coordinate extents, the coordinate system used, and more. For each scene, there are GeoTIFFs for each band. Since each band is stored in separate GeoTIFFs they will have to be indexed and stacked together for each scene.

Fortunately, the open source Open Data Cube software will handle all of the indexing, reprojection, and loading for this research study. It allows us to reproject our data into WGS 84 coordinates as well as index using latitude and longitude coordinates despite indexing them via satellite path and row.

Once the scenes have been coalesced into timeslice observations in the xarray dataset, it can be indexed by latitude, longitude, and time to return the requested band values for that pixel, which in this case is approximately a 30m area. Unfortunately, spectral reflectance instruments, unlike radar-based instruments, are significantly affected by assurance due to clouds. Clouds blocking the satellites view from the aperture of the sensing instrument will have to be parsed out from the scenes.

### 5.3 Sampling

While it would be ideal to train one algorithm for all Landsat data, that is not feasible nor is it a possible solution. The computation time and power would be astronomical and additionally, the errors generated from the model would be much higher than those compared to smaller regions. As a result, we randomly sampled smaller areas within the continental United States' Landsat data and engineered an algorithm for each area. A sample size of thirty smaller areas was generated, as the Central Limit Theorem states the average of the sample means will represent the population mean with a size of thirty or greater. As a result, this will give us an accurate measurement and potential improvement of the bias in the data.

To generate the samples, a shapefile containing the bounding polygon of the continental United States was used to verify the possible latitude and longitude values that were randomly generated. From the set of valid latitude and longitude values, we randomly selected thirty samples of 0.1 degrees latitude by 0.1 degrees longitude that fall within the original set. These coordinates were then converted into the corresponding satellite path and row coordinates in order to index the data for that location from the database. Additionally, all neighboring paths and rows were indexed as well to ensure all available pixels for the given coordinates would be retrieved.

The entirety of the last two years of data was loaded into memory and the first available timeslice with dirty pixels was labeled as the target timeslice. All preceding historical timeslices

necessary to fill the gaps left from masking out the dirty pixels are kept as well. Further details on how individual historical timeslices are used is available in Section 4.

# 6 Results

## 6.1 Neural Network Algorithm

A neural network algorithm was trained using 5-fold cross validation with a train-test split of 75/25. The 5-fold cross validation was chosen, so the computational time was not greatly increased since the datasets are very large. In order to obtain the best hyperparameters, a grid search was performed. The criteria for the best model was based upon the average MAE of all 30 models. The best model consisted of two hidden layers with a 20% dropout between each layer. The input neurons had a Lecun uniform initializer and a ReLu activation function. The first hidden layer consisted of fifteen neurons with a tanh activation function. The second hidden layer consisted of ten neurons and a ReLu activation function. Additionally, an Adam optimizer with a learning rate of 1e-5 was used. The loss curve of one model is shown in Figure 3.

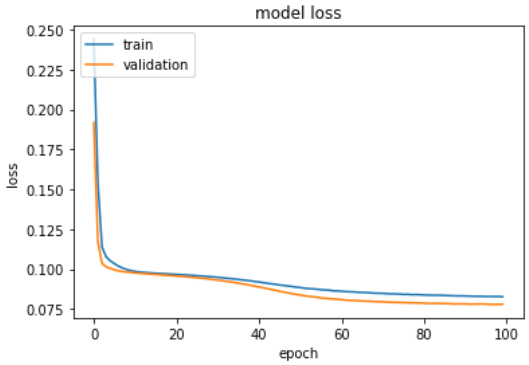


**Figure 3.** The loss curve of a specific area's neural network.

As shown in the loss curve, the neural network learned the relationship between the input and output rather quickly. The learning rate of the algorithm was reduced however, it did not have a significant effect on the loss.

The performance of the neural network model varied greatly between areas. If a neural network was tuned individually to each area, the performance would be greatly increased. However, this would greatly impact the time required to repair the data. As a result, the best overall performing model was chosen to predict each area. In future work, gathering additional data, such as NOAA climate data, might allow the neural network to learn more and therefore, perform better.

The neural network's hyperparameters were tuned by interpreting combined training and validation loss and accuracy plots. It should be noted however that it was quickly discovered

that the architecture of the Neural Network never generalized well to the varying timeslice samples. The model had a general performance that was comparable to the typical mosaicking method but was considerably slower.

### 6.2 Random Forest Algorithm

The Random Forest model was also trained using 5-fold cross-validation with a train test split of 75/25. The Random Forest was set up with a depth of 100 and an estimator count of 100. The values were obtained by performing a grid search algorithm using the MAE as the criteria for the optimal model. The rest of the parameters were left at their default values from the Scikit-learn library. One of the better results is shown below in Figure 4.

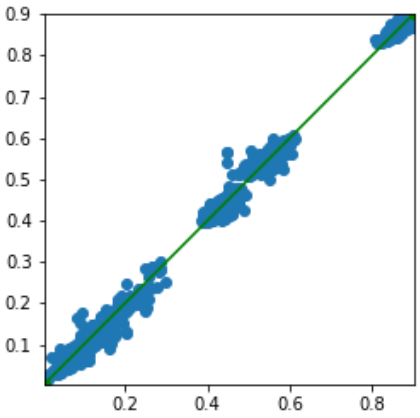


**Figure 4.** Predicted (y) vs Actual (x) normalized band values from a Random Forest Model.

As seen in Figure 4, the predicted band values lie fairly close to the line y=x. This indicated that the random forest model was successful at determining the relationship between the input bands and the output bands. The average MAE of the model was 0.0338 and had a standard deviation of 0.0305. On average, the Random Forest model was capable of a reduction of 80.1% error over the standard mosaicking method.

## 7 Analysis

### 7.1 Accuracy Comparison

For measuring the effectiveness of each model in the imputation of inaccurate data, the mean absolute error (MAE) metric was examined for each model. This metric was chosen in an effort to improve the interpretability and to quantify the success of each of the models' performances in relation to one another. The mean square error was also examined however, outliers were not a significant concern in the predictions as the goal of the models was to seek the best general fit.

In order to analyze the effectiveness of our algorithms overall, a baseline needed to be established in order to determine if our models were a viable solution. Since the purpose of the algorithm is to be applied iteratively to individual timeslices rather than across all timeslices simultaneously, it was decided that measurements be compared to the standard mosaicking method since it also is meant for the repair of a solitary slice in time. The most recently available pixel will be used where there is an absence of a clean pixel in the target timeslice. The most recent available pixels' ratios between their bands are likely to be the very similar, lighting aside, to the pixels that would have been present were they not obstructed from the instrument.

Taking the MAE of the mosaic is not possible for the affected regions of the data since the algorithm is not used on data unaffected by cloud coverage. As a result, the MAE must be estimated by randomly sampling the same number of pixels the model will validate on. The error will then be calculated between the corresponding pixels on the target timeslice, so the approximate error can be measured. This means the MAE is derived from pixels shared between the target timeslice and the historical timeslice currently being examined. If in the event a suitable number of shared pixels cannot be found, the next most recent timeslice will be used. However, in almost all cases, and in all of our test cases, the models will validate on several thousand shared pixels.

**Table 2.** Average Error summary statistics for the models.

| | RF | NN | Mosaic |
|---|---|---|---|
| **Standard Deviation** | 0.030325 | 0.432806 | 0.175605 |
| **Mean Absolute Error** | 0.035287 | 0.310498 | 0.229369 |

As seen in Table 2, between the Random Forest Regressor model and the neural network model, the random forest model performed much better than the neural network both the MAE and the MSE as shown in Figure 6. The random forest MAE on average was 0.0353 compared to 0.310 in the neural network models and 0.229 in the mosaic methodology. The overall distributions of MAE scores can be seen in Figure 6.

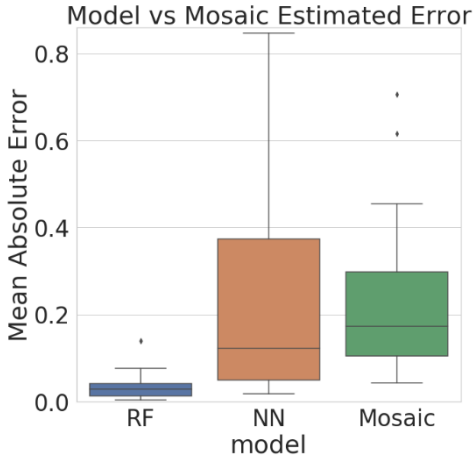


**Figure 6.** Distribution of the Mean Absolute Error between the models.

When plotting the red, green, blue values from the model predictions, the random forest generated much cleaner and more consistent images. The images shown in Figure 6 are images generated from two different samples in our framework. In the first sample, shown in the first column and first row, the patch generated from the mosaicking algorithm is much lighter than the surrounding pixels. It is easily observable to see where the cloud was present before the gap was filled. In the neural network generated image, shown in the first column and third row, the prediction is close to the average of clean pixels' bands. It is also easily observable to see where the cloud was present. Finally, in the random forest first sample prediction, the pixels generated are much less observable and thus, much more accurate to the true values.

In the second sample, the images generated from the three models are shown in the right column. The values predicted from all three models are much closer than in the previous sample. However, there are slight differences in the images. The predicted values in the upper middle section of the images in the neural network and the mosaic algorithm are slightly darker than in the random forest. When comparing the area to the original pixels, the random forest model more accurately represents the cloud covered area.

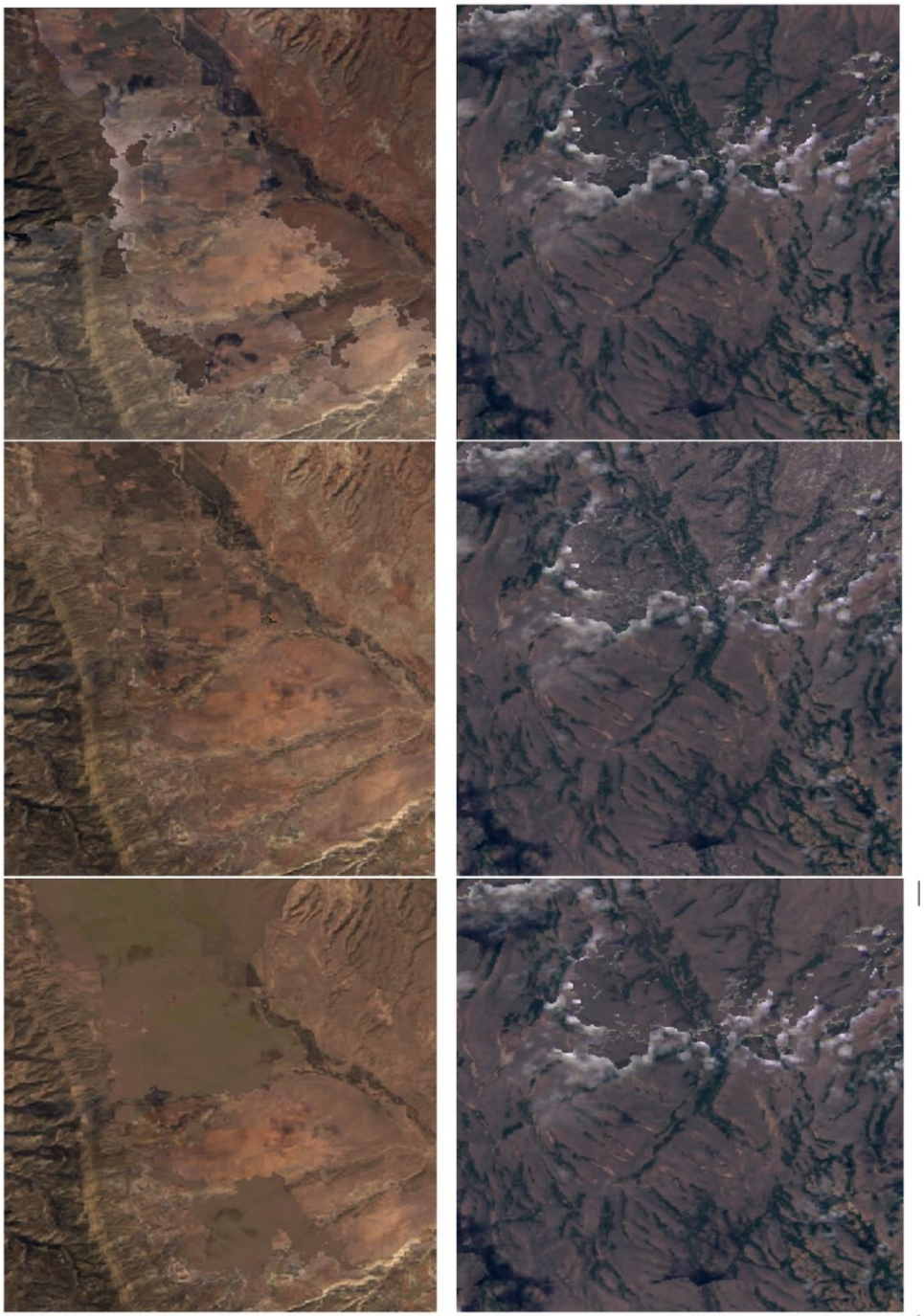

**Figure 6.** This grid shows the performance of the models with the transformations applied to the mosaic patch. The left column is one sample and the right is another sample. The top row is the standard mosaic, the middle row is the Random Forest Model, and the last row is the Neural Network.

### 7.2 Time Comparison

It is hard to determine the exact runtime of these algorithms due to a variety of factors. First, the training size varies greatly depending on the size of the swath of land chosen and the amount of shared clean pixels between the timeslices. Areas with fewer shared pixels between the two timeslices, likely due to cloud data, will train in about two minutes though it should be noted that larger training sizes might be preferred. However, the areas with lesser cloud coverage and several shared clean pixels between the timeslices may take as long as five minutes to train. Additionally, depending on the hardware performing these operations, speeds can vary greatly between systems.

If speed is more important than the accuracy of the patch to the extent that five additional minutes is not reasonable then the mosaicking algorithm is superior to both the random forest and neural network algorithms. The mosaic algorithm takes less than a second to run for in almost all cases. This is far superior compared to the random forest and neural network algorithms even with the leveraging of massive parallel processing. As a result, if band errors are not the central concern in a model, then the mosaic algorithm is a better option.

It should be noted again that the speed of this process was not a central focus since the algorithm is intended to be used as an additional preprocessing step done after ingestion and well before any analysis takes place. This would mean that there would be no need to mosaic during analysis since data could be repaired using the Random Forest algorithm proposed and stored for later use. Later when needed, the repaired timeslice could be loaded instead.

To be a viable suggestion for this application, the only requirement was that the algorithm is able to process an entire scene of Landsat data before a new one was generated. In order to keep the time reasonable, however, it was decided that the overall process shouldn't take any more than 5 minutes or so for 0.1 degrees latitude by 0.1 degrees longitude swaths of data.

## 8 Ethics

Due to the potential societal and scientific use cases from our model, ethics were a foremost component in the design and implementation of the model. Our model is built upon open source software and data, thus there are a few ethical concerns.

Our data comes from the USGS' Landsat data program. As mentioned above, the organization implemented a 'free-and-open data policy' to the public. Before this policy, the organization charged for downloads which provided a source of funding for the program. However, this policy was created to encourage the public to use the data for research and exploration across all industries, therefore the decision was justified with the increased number of peer-reviewed articles. Therefore, inhibiting resources to provide more accurate data would cause unnecessary public harm. The accuracy and errors associated with models are dependent on the integrity of the data. Our model has the potential to make models based upon it more accurate and in return, make more informed decisions. As a result, our model's hyperparameters will be open source for use available to the general public.

Consequently, with our model potentially being open source, this also produces ethical issues. While the model tends to be more accurate than current repair methods, there is still error associated with it.

Though the Open Datacube is free and open source, one of the authors of this study, Griffin Lane, is a previous employee of Analytical Mechanics Associates' NASA Open Datacube programming team and would be privy to proprietary information. In order to ensure that full transparency was achieved, a request stating the extent of the intended use of the software was relayed to Analytical Mechanics Associates. A response was received indicating that the use of the software was allowed as long as proper attribution was done, only publicly available sources were used, and no company resources were used. Nothing considered the property of NASA or AMA that was not publicly released with an open source license was used in this study. In order to ensure all parties are satisfied with the extent of the use of the software, a draft of this study was sent to AMA's NASA Open Datacube team.

## 9 Conclusion and Future Work

A methodology has been provided to repair cloud masking with Landsat satellite imagery. Our framework consists of collecting the data, identifying and extracting pixels with cloud coverage in a particular area, then training an algorithm on the clean pixels in order to reduce bias in the affected pixels. Out of the two algorithms analyzed, the Random Forest Regressor proved to most accurately repair the Landsat across the United States. The error was reduced significantly compared to current repair algorithms. However, future work is still needed.

While the random forest regressor was successful, it is not a viable solution when working with large regions of data. Our methodology was validated on small areas and therefore, cannot be applied to larger regions. While multiple algorithms can be used simultaneously to transform large regions, the computation power and time needed to train and predict would significantly increase. One possible solution to mitigate this problem could be to cluster areas then apply the one algorithm to each cluster. This would significantly reduce the number of models needed to transform a large area and thus, decreasing computational time and power.

Additionally, the Neural Network algorithm would benefit from a larger dataset. Neural networks tend to outperform other machine learning techniques when the data is significantly large. NOAA climactic data could be introduced to the model in order to increase training size. The climactic data would theoretically allow the neural network to pick up on the relationship between temperature and satellite instrument anomalies. Like the random forest algorithm, the computation size and power would be a significant concern. The hyperparameter tuning of the neural network model is much more tedious and hands-on than the random forest model, so a single model to incorporate the entire land area would be much more practical. However, the error would be much higher compared to a series of finely tuned models. While significant progress has been made, future work is still needed.

## References


1. Zhu Z., Wulder, D., Roy, Woodcock, C., Hansen, M., Radeloff, V., Healey, S., Schaaf, C., Hostert, P., Strobl, P., Pekel, J., Lymburner, L., Pahlevan, N., Scambos, T., "Benefits of the Free and Open Landsat Data Policy", Remote Sensing of Environment, Vol. 224. (2019) 382-385
2. "Landsat Band Designations". United States Geological Survey (USGS). Accessed 17 October 2018. https://landsat.usgs.gov/what-are-band-designations-landsat-satellites
3. Benger, S.N., "Remotely Sensed Determination of Flood Surface Gradients for Hydrological Modelling of Semi-Arid Floodplains," IEEE International Geoscience and Remote Sensing Symposium. Proceedings, Toulouse. (2003) 2950-2952.
4. "Using Landsat Satellite Imagery in Cancer Research". United States Geological Survey (USGS). Accessed 27 January 2019. https://eros.usgs.gov/sites/all/files/external/eros/science/cancerresearch.pdf
5. "Landsat Known Issues" United States Geological Survey (USGS). Accessed 17 October 2018. https://www.usgs.gov/land-resources/nli/landsat/landsat-known-issues
6. Storey, J., Scaramuzza P., Schmidt, G., Barsi, J., "Landsat 7 Scan Line Corrector-Off Gap-Filled Product Development", Land Remote Sensing. (2005) 23-27.
7. Lin C., Tsai, P., Lai, K., Chen, J., "Cloud Removal From Multitemporal Satellite Images Using Information Cloning," IEEE Transactions On Geoscience And Remote Sensing, Vol 51. (2013)
8. King, M., Platnick, S., Menzel, W., Ackerman, S., and Hubanks, P., "Spatial and Temporal Distribution of Clouds Observed by MODIS Onboard the Terra and Aqua Satellites," IEEE Transactions on Geoscience and Remote Sensing, Vol. 51. (2013) 3826-3852
9. Guo, Y., Li, F., Caccetta, P., Devereux, D., Berman, M., "Cloud Filtering for Landsat TM Satellite Images Using Multiple Temporal Mosaicing," 2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS). (2016) 7240-7243
10. Li, H., Wan, W., Fang, Y., Zhu, S., Chen, X., Liu, B., Hong, Y., "A Google Earth Engine-enabled Software for Efficiently Generating High-quality user-ready Landsat Mosaic Images", Environmental Modelling & Software, Vol. 112. (2019) 16-22.
11. Albayrak, A., Wei, J., Petrenko, M., Lynnes, C., Levy, R., "Global Bias Adjustment for MODIS Aerosol Optical Thickness using Neural Network," Land Remote Sensing. (2013)
12. Chander, G., Haque, M., Micijevic, E., Barsi, J., "Landsat 5 Thematic Mapper (TM) Recalibration Procedure for Data Processed using the National Landsat Archive Production System (NLAPS)," IGARSS 2008 - 2008 IEEE International Geoscience and Remote Sensing Symposium (2008) 1360-1363.