Contents

[1. Introduction 3](#_Toc32315728)

[2. Background 3](#_Toc32315729)

[2.1. Permafrost 3](#_Toc32315730)

[2.1.1. The Active Layer 3](#_Toc32315731)

[2.2. Ground Ice 4](#_Toc32315732)

[2.3. Thermokarst 4](#_Toc32315733)

[2.4. Alaska Thermokarst Model 4](#_Toc32315734)

[2.5. Random Forest Models 4](#_Toc32315735)

[3. Data 4](#_Toc32315736)

[3.1. Study Area and Period 4](#_Toc32315737)

[3.2. Air Temperature 5](#_Toc32315738)

[3.3. Precipitation 5](#_Toc32315739)

[3.4. Location 6](#_Toc32315740)

[3.5. Elevation, Slope, and Aspect 6](#_Toc32315741)

[3.6. Summary of Features 6](#_Toc32315742)

[4. Methods 7](#_Toc32315743)

[4.1. The Original Model 7](#_Toc32315744)

[4.2. Software 7](#_Toc32315745)

[4.3. Finding Baseline Hyperparameters 7](#_Toc32315746)

[4.4. Comparing Models 9](#_Toc32315747)

[4.5. Scenarios for Sensitivity Analysis 11](#_Toc32315748)

[5. Results 13](#_Toc32315749)

[5.1. The Baseline Random Forest 13](#_Toc32315750)

[5.2. Feature changes 13](#_Toc32315751)

[5.3. Hyperparameter changes 13](#_Toc32315752)

[5.4. A Final Model 13](#_Toc32315753)

[6. Discussion 13](#_Toc32315754)

[7. Future Work 13](#_Toc32315755)

[8. Conclusion 13](#_Toc32315756)

[9. References 13](#_Toc32315757)

# Introduction

# Background

# Permafrost

Permafrost is soil that is permanently frozen that occurs primarily in polar and alpine environments. Soils are considered to be continuously frozen if the mean soil temperature remains at or below 0 °C, the freezing point of water, for two or more consecutive years [1] [2] [3]. Permafrost has its greatest extent in areas where it has existed for thousands of years or longer. Permafrost generally occurs terrestrially, but also exists in the seabed in polar regions [1] [2]. Permafrost affects approximately 25% of the earths land area mostly in the arctic and sub-arctic [2] [3]. Permafrost covers approximately 80% of Alaska, 50% of Canada, and 60% of Russia [2].

Permafrost is classified in to three groups: continuous, discontinuous, and sporadic. Continuous permafrost covers 90% or more of a specific landscape with mean annual soil temperatures around -8 °C. Discontinuous permafrost occupies 50-90% of landscapes in permafrost areas with mean soil temperatures around -5 °C. Sporadic permafrost covers less than 50% of these environments, and the annual soil temperatures near 0 °C [2] [3]. Stable mean annually temperature is important to permafrost, as low heat flux is important for maintaining permafrost. Soil moisture, air temperature, snow cover, aspect, and elevation also influence the local heat fluxes. [2] [3]

Permafrost depth ranges from more than 1000m to only a few meters at its southern limits. The deepest know permafrost occurs in Siberia and is around 1400m thick [1] [2] [3]. The maximum depth of permafrost is limited by heat in the earth’s mantle [3]. Most permafrost is thousands to millions of years old, but some has formed more recently [1] [2].

# The Active Layer

During summer, as temperatures rise above 0 °C, the top layer of permafrost seasonally thaws. This seasonal thaw is known as the active layer [1] [2] [3]. The active layer depth ranges from a few decimeters in the far north to several meters in the discontinuous permafrost zone [1]. The active layer buffers permafrost from warm summer temperatures [2]. The conditions at the surface of the soil determine the influence the of air temperature on frozen soils. These conditions include slope, aspect, soil moisture, and snow cover [3]. Warmer summer temperatures are increasing the active layer depth and increasing destabilization of near surface permafrost [2] [3]. `

# Ground Ice

Ground ice is any ice in freezing or frozen soils, and occurs extensively in permafrost zones [3]. There are many types of ground ice ranging from massive ice deposits to small ice crystals in soil called pore ice. Ground ice can be ice that has formed in the soil or ice that has become covered as soil is deposited [2]. Ice wedge arrays are a type of ground ice that occur in a estimated 10% of the permafrost in Alaska’s arctic coastal plain [1].

Ice wedge ground ice occurs as moisture seeps into cracks in permafrost. These wedges grow as the ice freezes and thaws allowing more water in. These wedges reach 10m in depth and range from 2-3m wide [1]. Networks of ice wedge features come together in distinctive polygonal patterns called ice wedge polygons that range in size from 8 to 18 m across [1] [3]. Ice wedges, along with other ground ice features, are prone to completely melt as temperatures warm [2]. This leaves distinctive depressions in landscapes a process is called thermokarst [1].

# Thermokarst

# Alaska Thermokarst Model

# Random Forests

Random forest models are an ensemble machine learning technique that combines the results of many decision trees. Random forest models can be used as classification or regression tools. Each tree in a random forest is constructed using a randomized subset of the training data, and random subset of questions at each node. Final results are calculated by taking the mode( for classifcation), or mean (for regression) of all the trees. This corrects for possible overfitting that a single tree may have. The mechanics of each concept used in random forests are described in the following sections. (bams, and bermien).

# Decision Trees

Decision trees are a network of decision nodes and leaf nodes. Each decision node has exactly two children nodes which may be either another decision node or a leaf node. The split at each decision node is based on comparing the value of a single input feature traveling left or right based on whether the comparison is true or false. For a prediction, the tree is traversed for a set of inputs through each node, starting at the root, until a leaf node is reached. Leaf nodes contain the final result of the prediction, for classification this is a label while for regression this is a numerical value (bams, p3). [Add example]

# Bagging

Bagging is a method of Bootstrap aggravating. In general, Bootstrapping refers to selecting a random subset of samples from a given set of data with replacement. Giving a training set, size *n*, bagging builds modified training data, size *n*, for each tree by selecting n samples with replacement from the original training data. This method is called tree bagging. Feature bagging can also occur, and only selects a subset of features at each decision node. Using these bagging techniques results in largely uncorrelated trees.(bams, p3, b2, brimen).

# Appilcations

# Data

# Study Area and Period

The study area consists of Alaska’s arctic coastal plain. This area mostly consists of tundra north of the Brooks Range in Alaska. All data used is in the form of geotiff raster files to ensure consistency of the locations of each cell. The data have all been converted to a 1km square scale. This gives the rasters an area of 415 rows by 1096 columns. Add extenet bounds.

The climatological data exists for a period of 115 years starting in 1901. This is also the time period used to create the training data for the Random forest models described in section 4.3, and others. For the Random Forest models the data is subsampled into 25, 50, and 75 percent portions by taking random raster cells from across the full 115 year period. For data were there is no change, or very little, from year to year like geolocation the rasters are assumed to be constant over time. The raster projection used is the NAD83 Alaska Albers equal-area conic projection (EPSG:3338). ADD CITE(https://spatialreference.org/ref/epsg/nad83-alaska-albers/)

# Air Temperature

Air temperature data is used in a Degree Day format which represent heating or cooling. Here freezing Degree-days (FDD) represent winter temperatures while thawing degree days(TDD) represent summer temperatures. Degree Days are the sum of all daily average temperatures in given period in this case summer or winter. The Degree-day data used was derived from the Scenarios Network for Alaska and Arctic Planning (SNAP) Historical Monthly Temperature - 1 km CRU TS data which contain downscaled estimates of monthly mean temperatures raster data at a 1 km square resolution for 1901 to 2015 [4]. [5]Data for the ACP was clipped to the extent described in section 3.1.

To calculate FDD, and TDD from the monthly data using spline interpolation provided by Scipy. A univariate spline function was fit to the data, producing a periodic function with 2 roots per year, representing the change from summer to winter or winter to summer. This function was then integrated for all periods for all periods from root N to N+1 for all roots of the function in the given data range. This produced a set of alternating negative (FDD) and positive (TDD) values for each year. This process was done for each pixel in the Rasters. Any pixels where the method failed were interpolated from the surrounding pixels. This occurred in less than ten locations. FDD values were labeled according to the year where the winter starts. ADD FIGURES

# Precipitation

Precipitation data are in total millimeters and were derived from SNAP Historical Monthly Precipitation - 1 km CRU TS data which contain data downscaled to the same parameters as the SNAP air temperature data [5]. The monthly data are summed into the desired time periods used in our models. Values from April to September are summed for summer precipitation (SP) . Values for August and September are summed for late summer precipitation (LSP). October and November are summed for early winter precipitation (EWP) while October through March is summed for full winter precipitation (FWP).

# Location

Location data is derived from the raster data for FDD, though any georeferenced raster for the proper area of interest would work. The affine transform was used to calculate the latitude and longitude for the study area. These values were then saved as raster files themselves. The values in these raters represent latitude and longitude for the center of the pixel in NAD83 Alaska Albers equal-area conic projection (EPSG:3338).

# Elevation, Slope, and Aspect

Elevation data

# Summary of Features

Table 1 Summarizes the data used as features in the initial Random Forest models. The name of each feature is shown along with a shorter representation used throughout this paper. The features are climate related or physical location based. The source of the data is also listed.

Table : Summary of Features

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Representation** | **Type** | **Source** |
| Freezing Degree Day | FDD | Climate | Derived from SNAP monthly air temperature data |
| Thawing Degree Day (First Summer) | TDD | Climate | Derived from SNAP monthly air temperature data |
| Thawing Degree Day (Second Summer) | TDD+1 | Climate | Derived from SNAP monthly air temperature data |
| Summer Precipitation (First Summer) | SP | Climate | Derived from SNAP monthly precipitation data |
| Late Summer Precipitation (First Summer) | LSP | Climate | Derived from SNAP monthly precipitation data |
| Early Winter Precipitation | EWP | Climate | Derived from SNAP monthly precipitation data |
| Full Winter Precipitation | FWP | Climate | Derived from SNAP monthly precipitation data |
| Summer Precipitation (second Summer) | SP+1 | Climate | S Derived from SNAP monthly precipitation data |
| Latitude | Lat | Physical | Derived from raster affine transform |
| Longitude | Long | Physical | Derived from raster affine transform |
| Elevation | Elev | Physical | Derived from |
| Slope | Slope | Physical | Derived from elevation |
| Aspect | Aspect | Physical | Derived from elevation |

# Methods

# The Original Model

# Software

Model implementation and analysis were done in Python (version 3.7.3). The Conda package manager was used to create the Python environments used. Conda is the package manager used and created for the Anaconda platform. It allows for the creation of isolated python environments that can be shared between systems [5]. The environment file used is included in the Appendix.

The Random Forest regressor in Scikit-learn (version 0.20.3) was used to create the random forest models. This Random Forest implementation has many useful features including providing the feature importance, and ability to provide a decision path. There are also several hyperparameter available to control tree growth. If these parameters are unset their defaults create fully grown trees which can be very large [6].

# Finding Baseline Hyperparameters

Random Forest models have several hyperparameters that can be adjusted to affect the accuracy of the model being trained. Determining what values of these hyperparameters optimizes the model results can be difficult. Scikit-learn provides access to several hyperparameters. These include the number of trees, the maximum depth of each tree, and the maximum depth of each tree. Other parameters influence how the trees are split at each decision node. The Scikit-learn hyperparameters considered are summarized in Table 1.

Table : Summary of Random Forest Hyperparameters

|  |  |  |
| --- | --- | --- |
| **Hyperparameter** |  | **Description** |
| n\_estimators | e | Number of trees in the forest. |
| max\_depth | md | Maximum depth of each tree. |
| min\_samples\_split | mss | Number of samples required to split a node. |
| min\_samples\_leaf | msl | Number of samples required at each leaf. |
| max\_featuers | mf | Maximum number of features considered at each split. |
| max\_leaf\_nodes | mln | Maximum leaf nodes of each tree. |

In order to determine which combination of hyperparameters creates the best random forest model we looked at various combinations of parameters. The values of each hyperparameter considered are shown in Table 2. Also included in the table are the three percentages of the training data used to train the models. The combinations of these hyperparameters were examined in a brute force manor. Each variation of a hyperparameter was combined with each possibility of other the hyperparameters. With this method there were 5,184 models to train. {talk about the n\_E = 150 } ?

Table : Hyperparameter Values

|  |  |
| --- | --- |
| **Hyperparameter** | **Values** |
| n\_estimators | 10, 50, 100 |
| max\_depth | 12, 25, 60, 100 |
| min\_samples\_split | 2, 5, 10 |
| min\_samples\_leaf | 1, 2, 4, 8 |
| max\_featuers | AUTO, SQRT, LOG2 |
| max\_leaf\_nodes | 1,000, 5,000, 10,000, 50,000 |
| percent of training data | 25, 50, 75 |

Training the 5,184 possible models was accomplished using four systems with varying capabilities, and existing workloads. They ran either Mac OSX, Debian, or Ubuntu. Table 3 summarizes the systems used, and presents the time each ran the best model in. The Random Forest regressors were configured in a way that four parallel jobs were used to build trees in the forest. Statistics were collected on each of the models trained, but the models themselves were not saved at this time. The statistics recorded were the time to train each model, the time to predict the values for a given year, the mean difference and its variance, the mean absolute difference and its variance, and the median.

The work done by each computer was tracked via a csv file located in a git repository. This list initially contained the parameters to be used, a column to track the progress of the training, and empty columns for the desired statistics to be collected. An example of this format is included in the appendix. After each model was trained, the csv data was updated with the model’s statistics, and checked in with the git repository. During this step next set of Random Forest parameters was also acquired.

A problem arose with merge conflicts in the csv file after the first few days of running, and a step to correct this had to be developed. Merge conflicts were corrected by removing any duplicate rows, keeping completed rows over unrun or in progress rows with the same parameters, and deleting any of the auto generated merge conflict text. This step occurred right before the final resync that push new stats to the remote repository. This updated utility was deployed to only one of the systems, and the problem was corrected.

Table : Training Systems Summary

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Computer Name** | **OS** | **CPU** | **Memory** | **Time to run best model (4 jobs)** |
| Bristlecone | Ubuntu 18.04 | 3.2 GHz  Intel Core i7 (12) | 16 GB  1333 MHz  DDR3 | 0:37:14.363811 |
| Finwhale | Debian 10 | 3.2 GHz  Intel Core i7 (12) | 16 GB  1333 MHz  DDR3 | 0:35:21.714366 |
| Ocotal | Ubuntu 18.04 | 2.4 GHz  Intel Xeon (16) | 24 GB  1067 MHZ | 0:58:27.710421 |
| Chickadee | Mac OSX 10.12 | 2.5 GHz Intel Core i7 (8) | 16 GB  1600 MHz DDR3 |  |

The bulk training of the random forest models was run over the period of about a month before the training programs were externally stopped. A combined training time of 125 days 17:38:38 was completed during this period. The order for training took the number of estimators into account before any other hyperparameter. Therefor all 10 and 50 estimator models were run during the testing period while only ~83% of all 100 estimator models were completed before the training was stopped. Figures 1 and 2 show how the model training times vary color coded by number of estimators.

# Comparing Models

The accuracy of the models was measured two main statistics the mean difference (MD) from the original model, and the mean absolute difference (MAD) from the original model. Both of these values have to be used because the MD values all stay close to zero due to many positive and negative values being used to calculate the mean. The Mean Absolute Difference corrects this by taking the absolute value of the differences before calculating the mean there for giving a measure of how far off the random forest model is predicting. Figures 1 and 2 show this phenomenon graphically. In figure 1 the points plotted all show all numbers near zero while figure 2 shows numbers that

A screenshot of a computer

Description automatically generated

Figure : Mean Difference

A screen shot of a computer

Description automatically generated

Figure : Mean Absolute Difference

The best model was chosen by sorting the models by the mean of their absolute differences from the original model. This measure was used as it gives a better indication of how far off the Random Forest Model Predictions are. High accuracy model have low MAD values. A selection of top preforming models is presented in table (MADT). These models all had similar accuracy when applied to the study area for various years in the training period as shown in figure (FTOP). The ‘best’ model was selected because it had the lowest MAD value, other models in table (MADT) and figure (FTOP) were selected semi randomly. The hyperparameters from best model, now called the baseline Random Forest model (BRF), were used as the base of all the sensitivity analysis performed.

# Scenarios for Sensitivity Analysis

To determine the accuracy of the BRF, and to look at how small changes in hyperparameters effect Random Forest Accuracy several scenarios for sensitivity analysis were performed. These can be broken down in to two categories feature changes, and hyperparameter changes. The feature changes are designed to look at how the model results, and feature importance’s change if features with high or low importance are removed from the training feature set. This should help to verify their importance or not. The hyperparameter changes are designed to look at how small changes to the hyperparameters effect the model accuracy, as the method for choosing the BRF only looked at relatively large changes for many parameters. Several hyperparameters not examined for the BRF are also be examined. The scenarios for sensitivity are presented in

Table : Scenarios for Sensitivity Analysis

|  |  |  |
| --- | --- | --- |
| **Name** | **Type** | **Change** |
| OM as Feature | feature change | Add Original Model (training labels) as input feature |
| Random Data as Feature | feature change | Add a blob of random data as an as input feature |
| Remove Lat/Long | feature change | Remove latitude and longitude from input features |
| Remove top | feature change | Remove features that are indicated as most important |
| Remove bottom | feature change | Remove features that are indicated as least important |
| Vary max\_depth | Hyperparameter  change | Vary max\_depth around the value in BRF |
| Vary max\_leaf\_nodes | Hyperparameter  change | Vary max\_leaf\_nodes around the value in BRF |
| TBD |  |  |

# Results

# The Baseline Random Forest

# Feature changes

# Hyperparameter changes

# A Final Model

# Discussion

# Future Work

# Conclusion

# References

|  |  |
| --- | --- |
| [1] | N. Davis, Permafrost, Fairbanks, Alaska: University of Alaska Press, 2001. |
| [2] | W. Pollard, "Chapter 15 - Periglacial Processes in Glacial Environments," in  *Past Glacial Environments*, Second Edition ed., J. Menzies and J. J. van der Meer,  Eds., Elsevier, 2018, pp. 537-564. |
| [3] | T. Rowley, J. R. Giardino, R. Granados-Aguilar and J. D. Vitek, "Chapter 13 - P  eriglacial Processes and Landforms in the Critical Zone," *Developments in Earth*  *Surface Processes,* vol. 19, pp. 397-447, 2015. |
| [4] | Scenarios Network for Alaska and Arctic Planing, "Historical Monthly Temperature  - 1 km CRU TS," 2019. [Online]. Available: http://ckan.snap.uaf.edu/dataset/  historical-monthly-temperature-1-km-cru-ts. [Accessed 04 11 2019]. |
| [5] | "Conda," [Online]. Available: https://conda.io/projects/conda/en/latest/.  [Accessed 05 02 2019]. |
| [6] | scikit-learn-developers, "3.2.4.3.2. sklearn.ensemble.RandomForestRegressor -  scikit-learn 0.21.3 documentation," [Online]. Available: https://scikit-learn.org/0.21/modules/generated/sklearn.ensemble.RandomForestRegressor.html  #sklearn.ensemble.RandomForestRegressor. [Accessed 05 02 2020]. |