# Estimating the Snow water equivalent in the Tuolumne Basin using Random forests and Multi-Layer Perceptrons

George Cowie

December 17, 2020

**Abstract**

In this project i have used a Random forest and Multi-Layer perceptron regressor to model the Snow Water Equivalent in the Toulumne basin in the Californian Sierra Nevada. Accurate models of Snow Water equivalent are vital for water resource management in the earths many mountainous regions, but direct measurements remain sparse. The models were generated using static terrain parameters and snow depth measurements from NASA's Airborne Snow Observatory. The Random Forest model was optimised for the hyperparameters: number of trees and maximum tree depth, while the Multi-Layer perceptron model was optimised for: number of hidden neurons, learning rate and number of epochs. The random forest model was able to capture the spatial distribution of the Snow Water Equivalent and had a MSE of around 0.05m in years of moderate snow cover. It did not however manage to accurately model smaller areas of deep snow, and had a larger error for years with a deep snow pack. The Multi-Layer perceptron model performed poorly in all cases and could not accurately model the snow distribution. This was likely caused by sub optimal hyperparameters as other similar studies have found the two methods to give similar results.

## 1 Introduction

In mountainous regions snow cover dominates the regional hydrology and worldwide over 1.5 billion people rely on snow melt water for their water supply [16]. In the western United States $> 75\%$ of the freshwater originates from snow melt runoff. In this region the combination of increasing demand for fresh water from a growing population and changes in runoff volume due to climate change are creating serious water security challenges. With ever increasing $CO_2$ contentraitions in the atmosphere and growing urban areas these problems will only continue to become more severe [13].

Increasing temperatures are causing a reduction in seasonal duration of mountain

snowpacks and an increase of dust load from desert systems. Dust loading on mountain snowpacks reduces the albedo, thereby inducing faster melt [13]. Two of the most important factors in determining the seasonal melt runoff are the snow albedo and the snow water equivalen (SWE). However, these properties remain poorly quantified globally [13].
This data gap has motivated the development of the Airborne Snow Observatory (ASO). The measurments of the ASO are taken by planes flying at high altitudes, equpt with a lidar and image spectrometer, measuring snow depth, SWE and albedo. With the ASO both the spatial and temporal scale of these kind of measurements has been greatly expanded.[15]

Though good ASO covereage exists for some important basins in the United States, large portions of the earths mountainous regions remain unmapped. The massive area covered by snow globally makes it unfeasible to generate adequate data for accurate mapping of the SWE globally. This is where machine learning algorithms can play an important role. A number of properties of mountainous regions, like slope, elevation, canopy cover, can be accurately measured by satellites, giving good global coverage. Given that these properties are correlated to the snow cover is should be possible to train a machine learning algorithm with these predicators with the ASO snow depth as ground truth.

In this project i will be using two different regression methods to estimate the SWE, Random Forest and Multi-Layer perceptron. The algorithms will be trained and tested on available ASO data and static terrain parameters.

The report is structured into the following sections: theory, Data and Methods, Results, and discussion. In the theory section i take a quick look at the theory behind the regression methods. In the methods section i will go through the implementation of the regression methods and how the code is structured. In the results i will show how the models performed compared to the true measurements and some test metrics. Lastly i will discuss the results and compare the two methods.

## 2 Theory

In this section i will outline the theory behind the two regression algorithms used in this project. and the metrics used to evaluate them.

### 2.1 SWE

The target variable in this report is the SWE depth. This is the depth of water you would get from melting all the snow found in a given area, calculated as follows:

$$SWE = z \cdot \frac{\rho_{snow}}{\rho_{water}}$$

Where z is the snow depth, and $\rho_{snow}$ is the density of snow and $\rho_{water}$ is the density of water. For an explanation of how the snow depth and snow density are determined by the ASO see [13].

## 2.2 Random Forest Regression

Random forests are based on a group of decision trees, each trained on a different random subset of a training set. Then predicting the class that gets most votes. This is an example of a so called ensemble method [9]. Since Random Forests are an expansion of decision trees we will first have a look at the latter.

Decision trees are supervised learning algorithms that can be used for both classification and regression. At its very simplest a decision tree is a method that finds the features that contain most information about our desired target feature and splits datasets along these values, in order to end up with an underlying dataset that is as pure as possible.

Decision trees are composed of a series of nodes: root node, interior nodes and leaf nodes. The nodes are connected by so called branches. The leaf nodes contain the predictions of an instance, the interior nodes specifies the test of an attribute of the instance and the root node is the first attribute test from where the the test moves into the interior nodes and out to the leaves. [12]

The decisions at a node of a decision tree can be either numeric or a classification or a combination of these. In the case of this project the nodes will be numeric since we are using the tree for regression purposes.

In this project i have been using the Scikit-Learn functionality which utilises the Classification And Regression Tree (CART) algorithm [4]. The CART algorithm performs the data split by making two subsets using a single feature k and a threshold $t_k$. The quantities k and $t_k$ are chosen by searching for the purest subset, according to for example the gini factor G. The cost function that is to be minimised is [11]

$$C(k, t_k) = \frac{m_l eft}{m} G_{left} + \frac{m_r ight}{m} G_{right} \tag{1}$$

At the root node of the tree we place the subset that gives the purest value. This node will then make decisions, splitting the data into new branches. These branches lead to the next layer of interior nodes, further splitting the data. The same purity test is performed on the remaining data and the next nodes are chosen by subset that is purest. From this node we create new branches and until all the subsets have been used. If a decision results in a node that has the lowest score overall, this becomes a leaf node and no more branches are created. The CART algorithm has been used extensively for snow related predictions [6].

Trees have one aspect that prevents them from being the ideal tool for predictive learning, namely inaccuracy. They seldom provide predictive accuracy comparable to the best that can be achieved with the data at hand[10]. This makes Decision Trees very good with the data at hand, but bad at making predictions and prone to overfitting.

Random Forests are bagged decision trees and mitigate the deficiencies of the single decision tree by growing multiple trees by randomly sampling the subsets of predictors, creating a a weighted ensemble of trees. This method reduce overfitting and increases the accuracy of predictions [6].

## 2.3 Multi-Layer Perceptron

A Multi-Layer Perceptron (MLP) is a neural network composed of an input layer, one or more hidden layers and an output layer. Each layer is composed of artificial neurons called linear threshold units (LTU). The input and output of a perceptron are numbers and each input connection is associated with a weight. Every layer is fully connected to the next layer and includes a bias neuron [9].

The MLP is trained using a the backpropagation algorithm. The steps of the backpropagation algorithm are as follows:

1. Training instances are fed into the network
2. Predictions are made and error measured
3. Go through each layer in reverse and measure error contribution from each connection
4. Tweak connection weights to reduce error

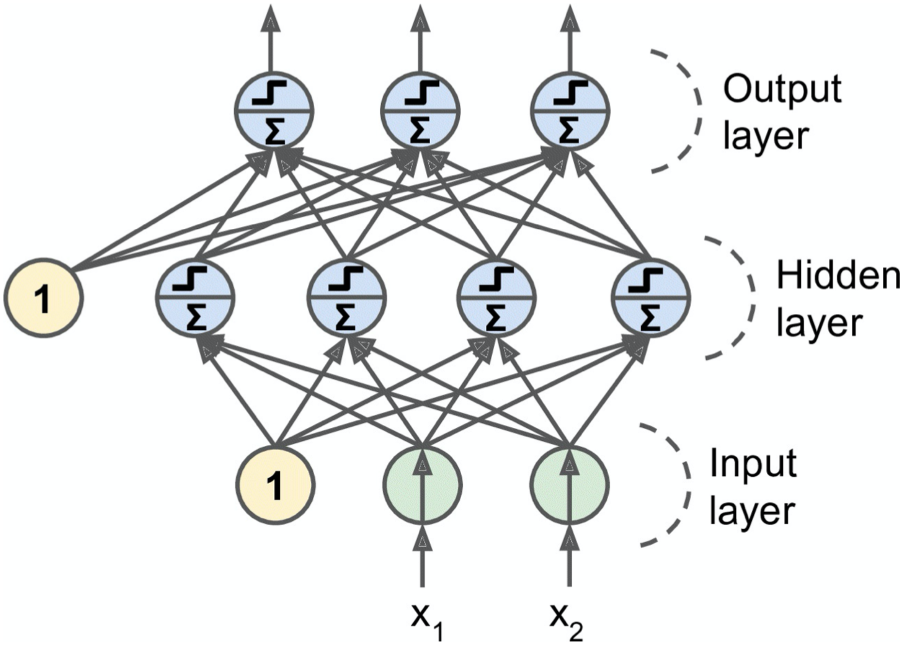

Figure 1: Multi layer perceptron from page 350 in [9]

### 2.4 Metrics

I have mainly used two test metrics to evaluate model performance, Mean squared error (MSE) and R$^2$.

The mean squared error is a risk metric of the expected value of the squared error between the true value $y_i$ and predicted value $\hat{y}_i$

$$MSE(y, \hat{y}) = \frac{1}{n} \sum_{i=0}^{n-1} (y_i - \hat{y}_i)^2 \tag{2}$$

The closer the MSE is to 0, the better the fit.

The R$^2$ score, also called the coefficient of determination, measures how well the true values are predicted by the model. The best possible score is 1.

$$R^2(y, \hat{y}) = 1 - \frac{\sum_{i=0}^{n-1} (y_i - \hat{y}_i)^2}{\sum_{i=0}^{n-1} (y_i - \bar{y})^2} \tag{3}$$

# 3 Data and methods

## 3.1 Data

The data for this project comes from the Tuolumne basin in the Californian Sierra Nevada. The basin covers an area of 5,070$km^2$. The basin is physiographically characterised by mountains,forest, plains, lakes and glaciers. A map of the study area can be seen in Figure 2. The map does not show the entire area covered in this project, but places the study area into a nice geographic context. The data contains 16 SWE depth maps from ASO lidar scans taken in the period 2016-2019 and static terrain parameters from the SRTM digital elevation model [8] and the National Land Cover Database [17]. The file was compiled by Kristoffer Aalstad

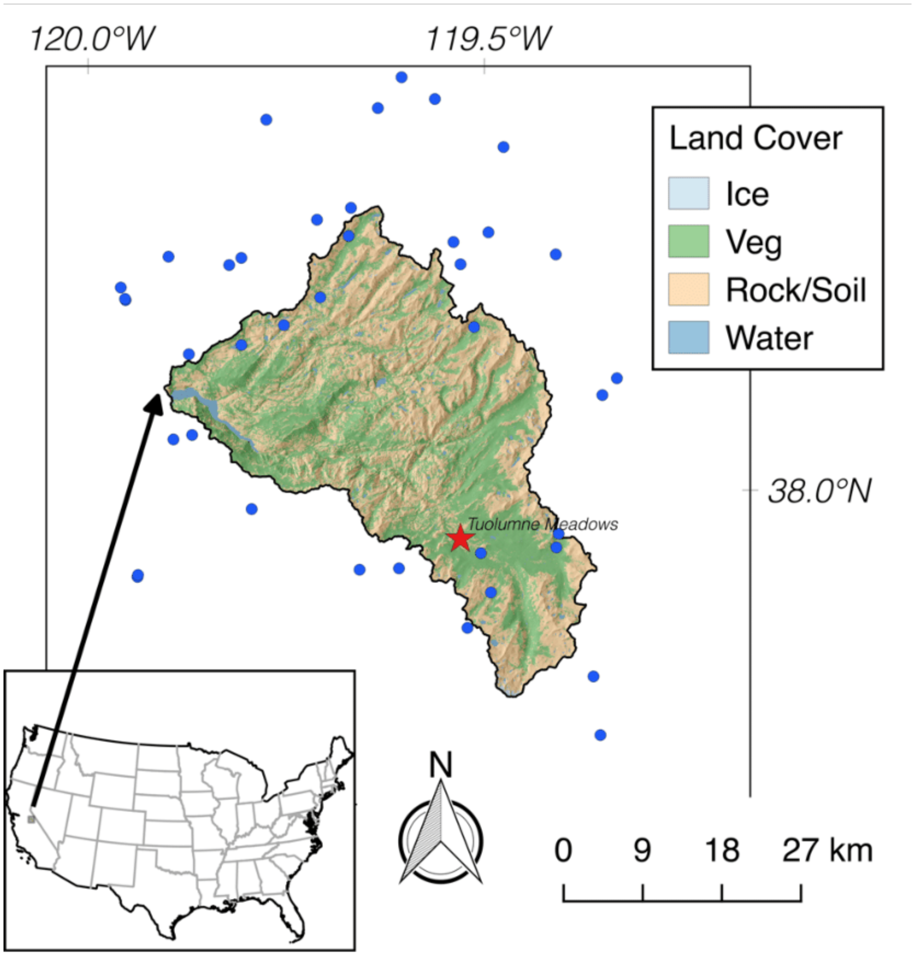

Figure 2: Map of the Tuolumne basin and its position in the United States [15]. My study includes a little more area to the north west of this map.

from the department of geoscience at UIO. Each flight has a year and day of year attached. When i refer to individual flights i will refer to them by the day of year and the year, as an example: 92/2016. This is the 92 day out of 365 in 2016.

The pixels are regularly spaced in zone 11s in the UTM projection, WGS84 ellipsoid, with the first pixel in the upper left corner of the image (i = 0,j = 0). The spatial resolution of the data is 100 meters per pixel.

The static terrain features and their labels are:
**Elevation [m]: 'z'**
**Aspect [°]: 'asp'**
**Slope [°]: 'slp'**
**Sky view factor: 'svf'**
**Canopy cover: 'CC'**

From the aspect it is possible to generate two predicators, namely the "northness"=-cos(asp) and "eastness"=sin(asp) of a slope. The Sky View Factor (SVF) is the fraction of the sky that is observable centred on a point, a point deep in a valley will for instance have a lower SVF than a point on a mountain peak. The other predictors are perhaps more self explanatory.

## 3.2 Methods

In this project i have used the regression functions available through Scikit-Learn for both Random forest and MLP [14].

### 3.2.1 Random Forest

For the Random Forest regression i have used the regressor: "sklearn.ensemble.RandomForestRegressor"[3]. The regressor has multiple hyperparameters that can be tuned. In my investigation i chose to limit myself to tuning the number of trees and the maximum depth of each tree. The tuning of both the number of trees and the depth was performed on the same subset of data. The subset i used was drawn from the flight on the date 92/2016. I used 20% of the data for training and had 80% reserved for testing.

To tune the number of trees i ran the regression model in a loop, iteratively increasing the number of trees for each loop and storing the results. The loop was run between 1 and 50 trees.

Tuning the maximum depth was done in a similar manner, iteratively increasing the maximum depth for each pass and storing the results. The loop was run between 1 and 50.

Next i evaluated the results using the $R^2$ [2] score, the MSE [1], bias and variance using the functions available from sklearn.

After tuning i ran the regression for each of the 16 ASO flights individually using the optimised hyperparameters. Again i looked at the $R^2$ score and the MSE for each flight. I also wished to view how the model performed under differing SWE conditions so i included a plot of the pixel sum of the SWE for each flight.

Finally the results are mapped back to the pixel coordinates and visualised.

### 3.2.2 Multi-Layer Perceptron

For the MLP regression i used the regressor: "sklearn.neural_network.MLPRegressor" [5]. This regressor has near infinite ways of tuning the hyper parameters, but i have confined myself to tuning the number of hidden neurons in each layer, the number of epochs and the learning rate. The method for tuning these parameters is identical to the one described for the random forest. For the hidden neurons i tested between 10 and 200 with an interval of 10, the learning rate was tested between $10^-7$ and 1 and the number of epochs were tested between 50 and 1000 with an interval of 50.

# 4 Results

The results section is split into the results for the Random forest and the MLP regression. In both cases i will first present the optimisation results, and then the results for every ASO flight. The optimisation was in both cases done on selected flights in order to save time as running the regression for all flights is very time consuming.

## 4.1 Random forest

In figure 3 we see the R2 score and MSE from runs with an increasing number of trees. Both curves have an initial steep slope, until the slope tapers off an increases asymptotically. The model accuracy gained from adding an additional tree is greatly reduced as we add more trees. I made a decision that 20 trees would be sufficient for the purposes of my project, as the accuracy does not increase significantly after this point and it gives a reasonable computational time.

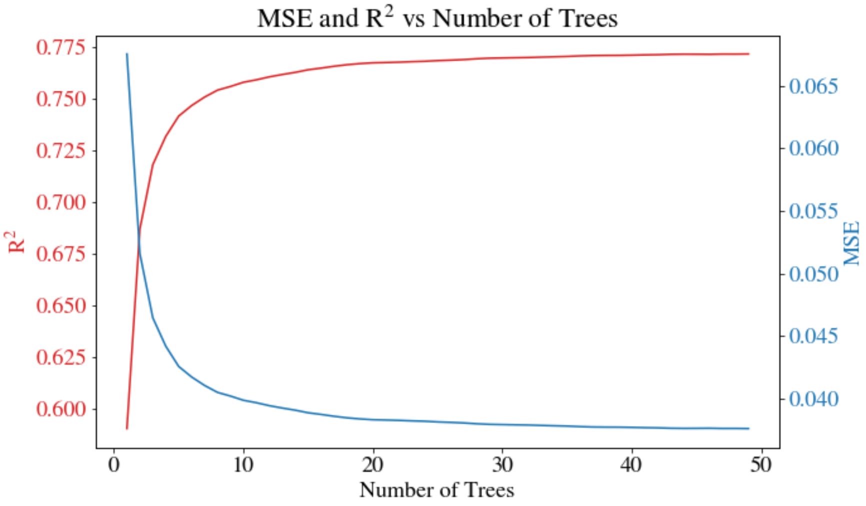

Figure 3: $R^2$ and MSE against number of trees. The left Y axis shows and red line show $R^2$ score and the right axis and blue line show the MSE. These values come from the first flight of the ASO in the data 92/2016. It was trained using a maxmum tree depth of 15.

In figure 4 we see how the bias and variance are impacted by changing the number of trees. The Bias remains almost constant, while the variance decreases while adding more trees. The reduction in variance by adding a new tree is high when there are few trees, but the improvement per new tree added decreases as the forest grows larger.

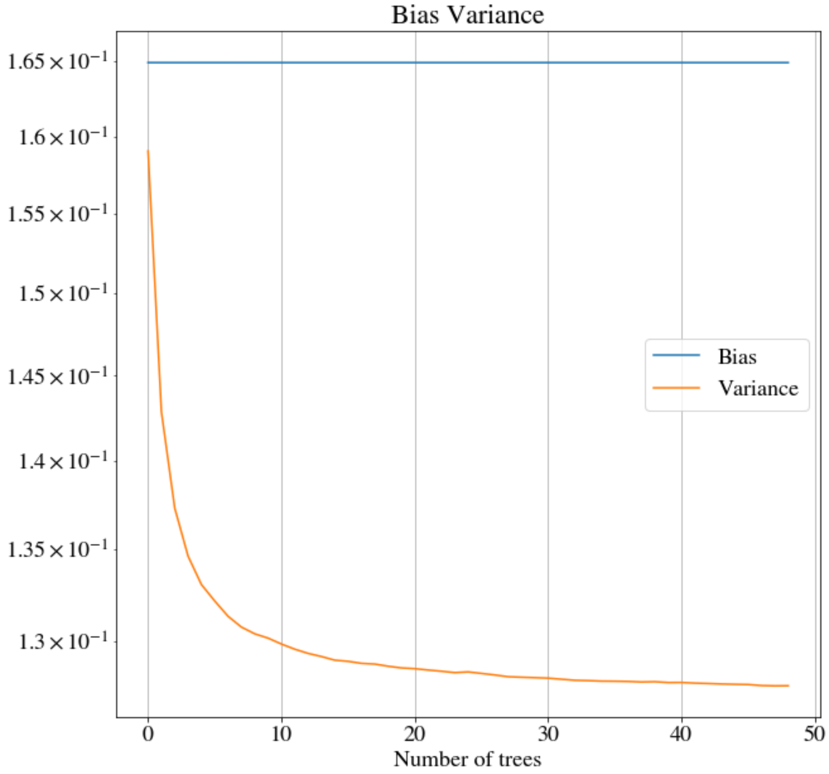

Figure 4: Bias and variance against the number trees. These values come from the first flight of the ASO in the data 92/2016. It was trained using a maxmum tree depth of 15.

In figure 5 we see the R$^2$ score and MSE as a function of the maximum depth of each tree in the random forest. The MSE decreases with added depth from 1 to when the MSE has a minimum value at a maximum tree depth pf 16, after this the MSE begins very slowly increasing again. I therefore chose 16 as the optimum depth of the tree. The same behaviour is shown by the R$^2$ score, except that it is inverse of the MSE curve.

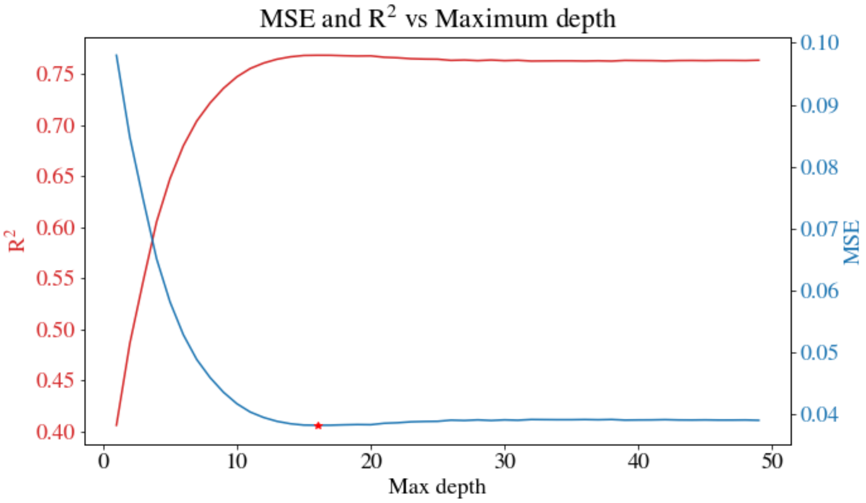

Figure 5: R$^2$ and MSE against depth. These values come from the first flight of the ASO in the data 92/2016. It was trained using 20 trees. The red star on the MSE curve indicates the minimum MSE value.

In 6 we see the bias and variance as a function of the maximum depth of the trees, as well as the test MSE. Again the bias remains near constant, but the variance increases with added depth. Although the variance is increased the test MSE is reduced, indicating that the increased variance does not have a dominating impact on the overall MSE.

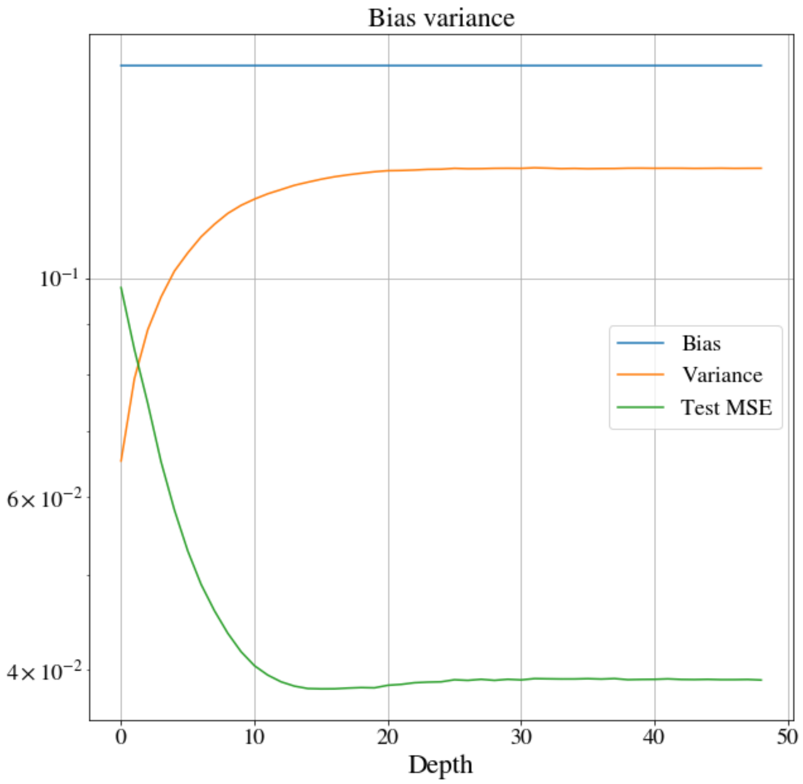

Figure 6: Bias and variance against depth. These values come from the first flight of the ASO in the data 92/2016. It was trained using 20 trees. The Y axis is on a logarithmic scale

In 7 We see the R$^2$ score and the MSE for each ASO flight. This should be viewed together with figure 8 that shows the sum of the SWE pixels in the entire basin and the MSE. Both R$^2$ and MSE show variations in for each pass, the variation seems to have a connection to the total SWE in the basin. There are two extremes that are particularly noteworthy. For the year with the lowest totalt SWE (228/2017) we see both a low MSE and a low R$^2$ score and in the year with the highest total SWE (107/2019) has the highest MSE and a R$^2$ score of 0.65. A moderate to high SWE consistently gave the best $R^2$ score.

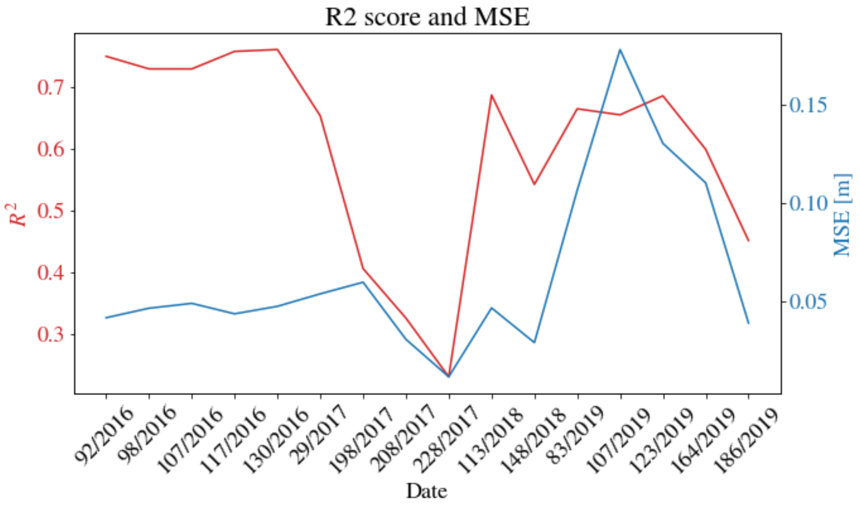

Figure 7: $R^2$ score and MSE for each ASO flight with the Random Forest. On the X-axis we have the date of each flight and on the twin Y-axes we have the $R^2$ score on the left and the MSE on the right. The results were generated using a forest of 20 trees with a maximum depth of 16.

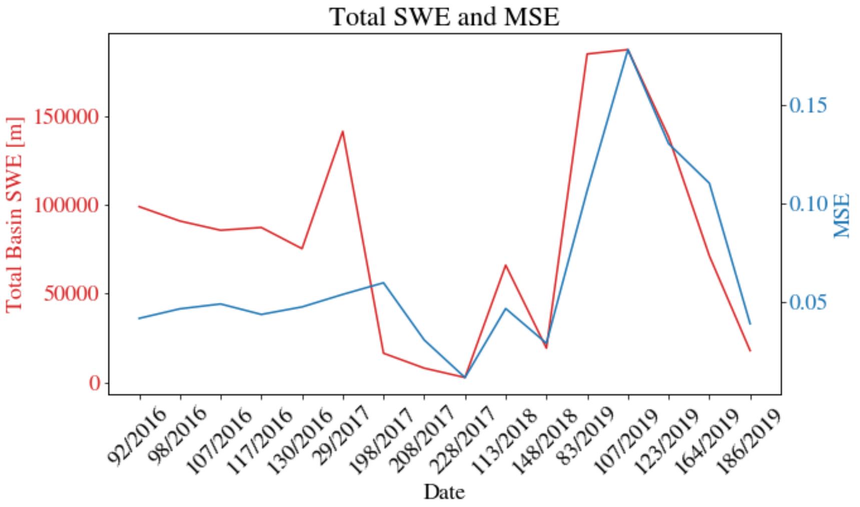

Figure 8: Total SWE and MSE for each ASO flight with the Random Forest

Figure 9, 10 and 11 are selected runs showing the SWE measured from the ASO and the emulated SWE from the random forest regression. The maps are created by training on 20% of the data. I chose to show these three passes since they display three different SWE conditions within the basin, figures 10 shows a low total SWE, figures 11 show the highest total SWE while figures 9 shows an SWE roughly between the two. The total snow cover for each pass is plotted together with the MSE in figure 8. From a purely qualitative point of view the regression is able to capture certain details of the snow cover rather well, i.e where there is no snow cover and where the snow is deepest. However, the absolute error shows a tendency to be large in areas where the ASO measured deep snow. This is particularly apparent in figure 11 where there was a high SWE across the whole basin.

Note that there are certain areas that have a SWE higher than the 1.5m on the colorbar. These areas are however few and far between so limiting the colorbar to 1.5m gives a better visual representation of the SWE.

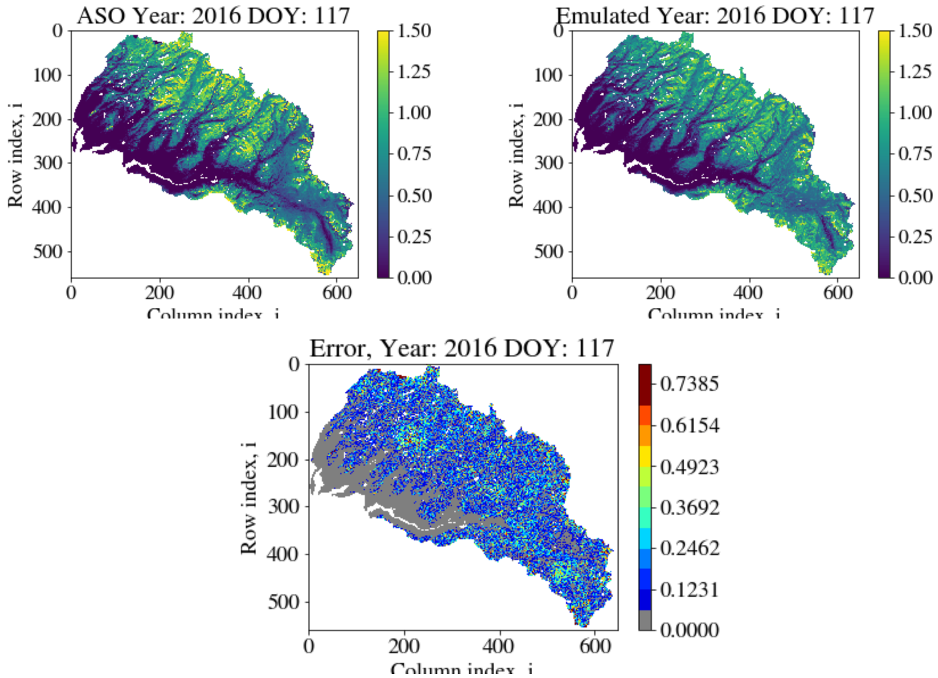

Figure 9: Maps of SWE measured by ASO, emulated SWE using random forest and the absolute error between the emulated and measured SWE.

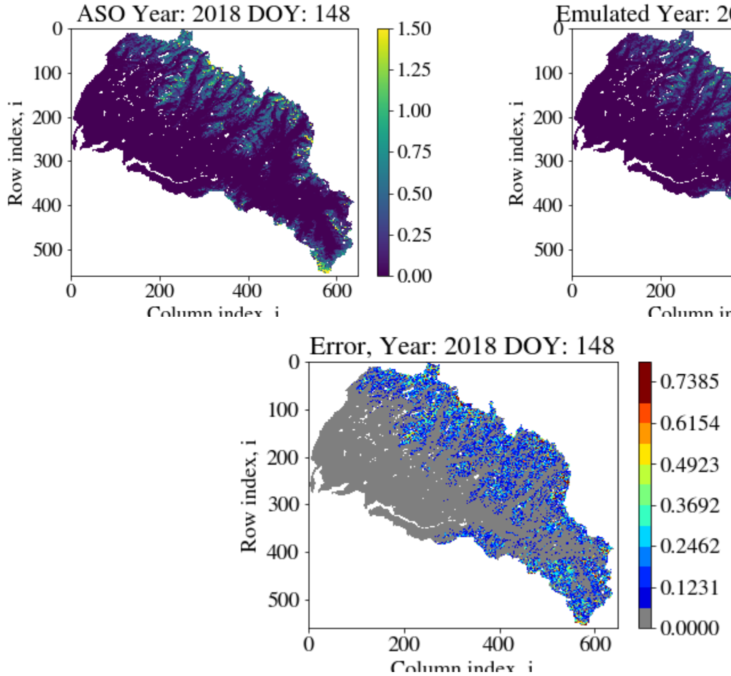

Figure 10: Maps of SWE measured by ASO, emulated SWE using random forest and the absolute error between the emulated and measured SWE.

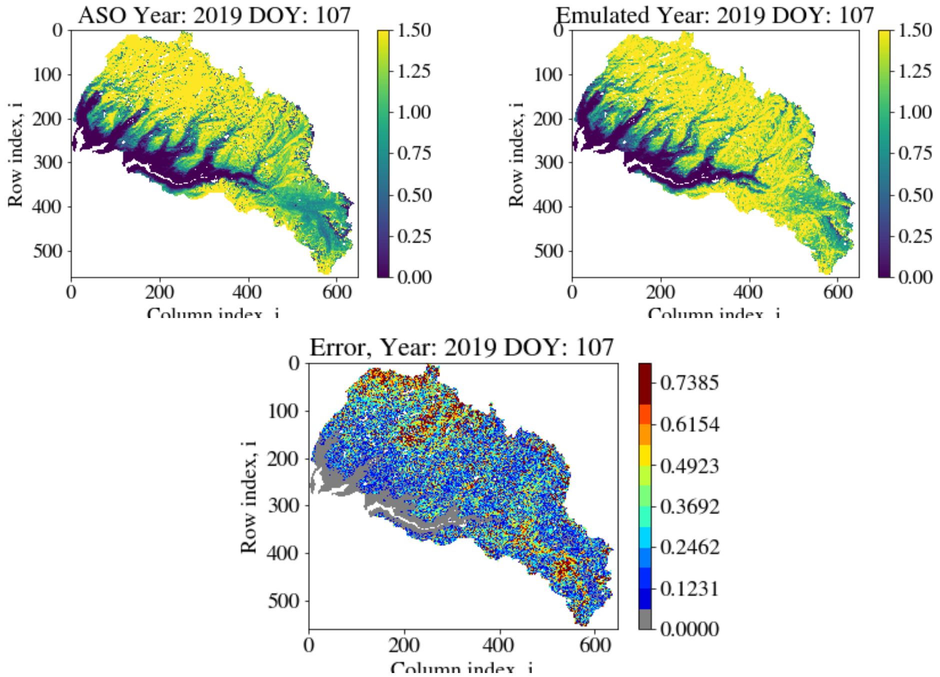

Figure 11: Maps of SWE measured by ASO, emulated SWE using random forest and the absolute error between the emulated and measured SWE.

## 4.2 Multi-Layer Perceptron

When optimizing the hyperparameters of the MLP model i chose a slightly different approach to what i did with the random forest. Since i observed significantly different MSE values for the maximum and minimum SWE using the Random Forest i chose to try optimizing the MLP using these extreme cases, in the hope of creating a model that could perform well in these extreme cases. The hyperparameters i tested were the number of hidden neurons in each layer, learning rate and number of epochs. The scale of the MSE is quite different in the minimum and maximum case so this should be in the back of ones mind when viewing these results.

In figure 12 we see the MSE against the number of hidden neurons. How the MSE changes in the case of the maximum and minimum SWE is quite different. For the maximum the MSE seems to stabilize around 0.28 after 50 neurons are added. The MSE in the case of the minimum snow cover seems more erratic, but is in fact consistently much smaller than in the case of the maximum, with an MSE ranging between 0.015 and 0.0158. The error in the minimum snow cover is 18 times smaller than for the maximum. I decided to choose the number of hidden neurons based on only the Maximum SWE data since there was no clear decision to be made based on the minimum. The minimum MSE value for the MLP trained on the maximum

SWE data occurs at 60 hidden neurons.

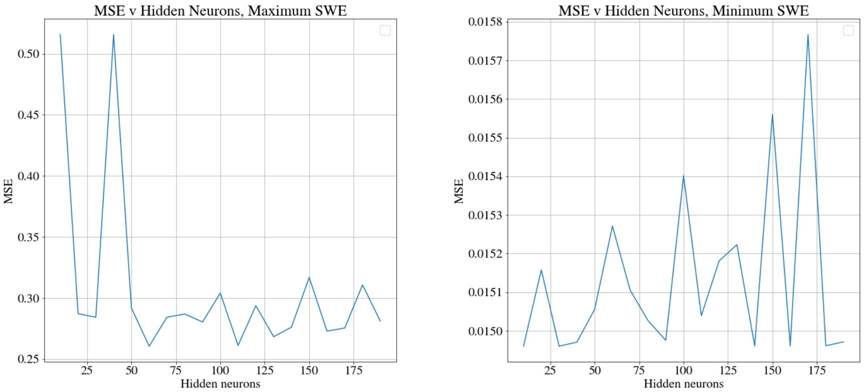

Figure 12: Plot of MSE against the number of hidden neurons in each layer with a learning rate of $10^4$ and 200 epochs. The left image is trained on the ASO pass with the maximum total SWE and the right with the minimum SWE. 20% of the data was used for training and 80% reserved for testing.

In figure 13 we see how the MSE changes with the learning rate. Again we see that there is no clear pattern in the case of the minimum snow cover, except for some spikes in the MSE between $10^{-7}$ an $10^{-6}$. I therefore chose the learning rate based on the MSE in the maximum case. A learning rate of $10^-4$ gave good results in both cases and is what i used for the final model.

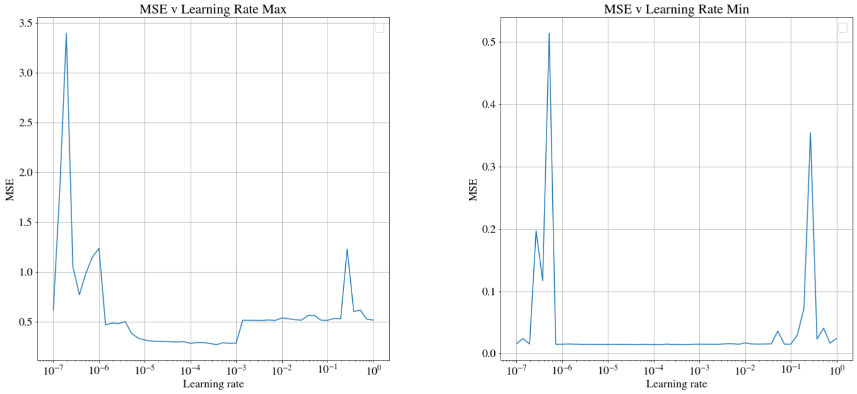

Figure 13: Plot of MSE against the learning rate using 60 hidden neurons and 200 epochs

In figure 14 we see the MSE as a function of the number of epochs. For the maximum SWE case the MSE generally became smaller with an increase in the number of epochs, though it oscillates quite a bit. With the Minimum SWE there is no clear pattern of MSE reduction with an increase in epochs. The areas of lowest MSE in the maximum case corresonds to an area of higher MSE in the minimum case. I decided to strike a balance between the two cases and the cpu time ti took to compute, and choose 250 epochs for the final model. In any case the change in MSE caused by added epochs is small compared to that of the learning rate and the number of hidden neurons, so this is perhaps not the most important hyperparameter.

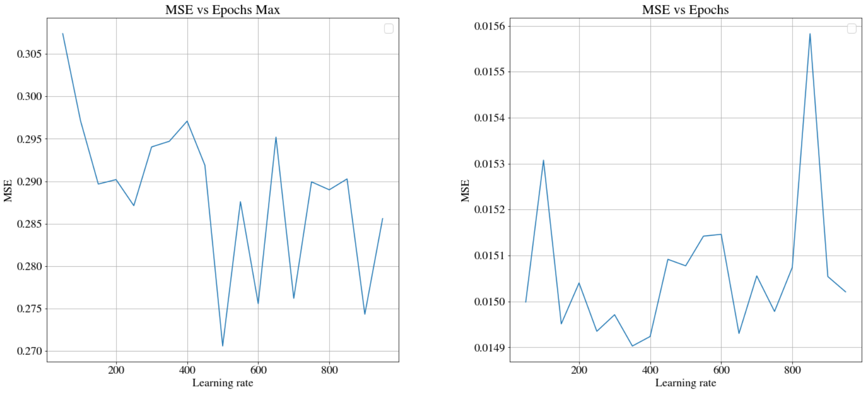

Figure 14: Plot of MSE against the number of epochs with 60 hidden neurons in each layer and a learning rate of $10^{-4}$

In figure 15 we see the R$^2$ score and MSE for each flight of the ASO. The shape of the curves are almost identical to that found using Random Forest, although the $R^2$ score and MSE are worse across the board.

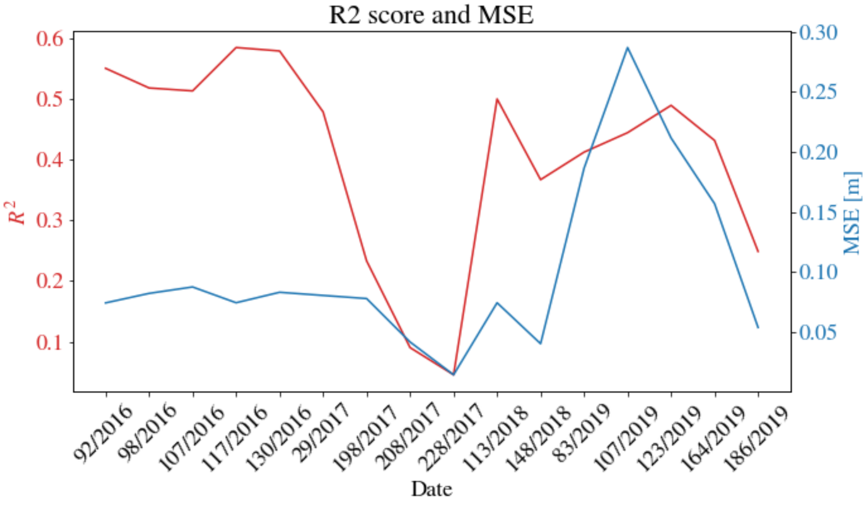

Figure 15: $R^2$ score and MSE for each ASO flight using the Neural Network.

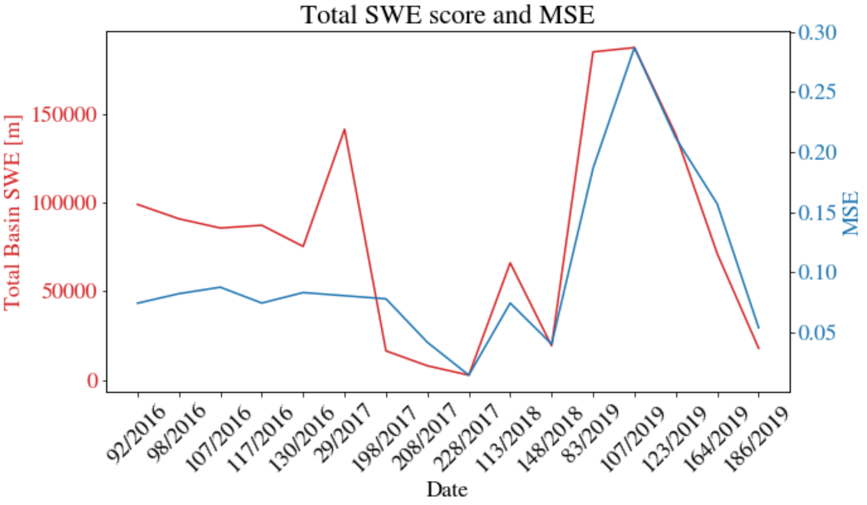

Figure 16: MSE and total SWE in the basin for each ASO flight using 60 hidden neurons, 250 epochs and a learning rate of $10^{-4}$

Figure 17, 18 and 19 are selected runs showing the SWE measured from the ASO and the emulated SWE from the MLP regression. These are the same flights used for the Random Forest. Viewing these plots it is obvious that the model is not performing well. The SWE distribution not correct, showing snow where there is none and not accurately predicting the depth. Almost none of the finer details of the true SWE are captured. The error is most apparent in figure 19 where the modelled results show the highest SWE concentrations in the south east side of the basin, while the highest SWE concentration is in fact located in the north west side.

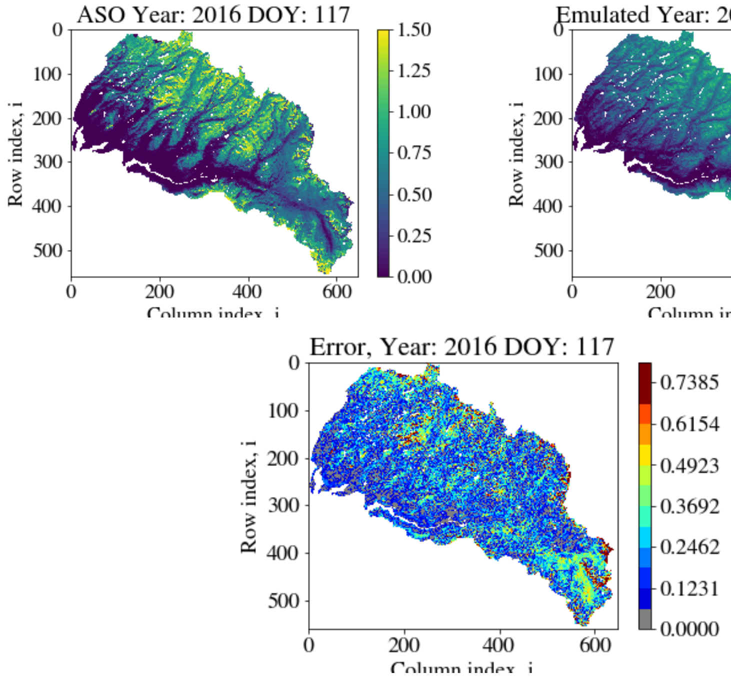

Figure 17: Maps of SWE measured by ASO, emulated SWE using MLP and the absolute error between the emulated and measured SWE.

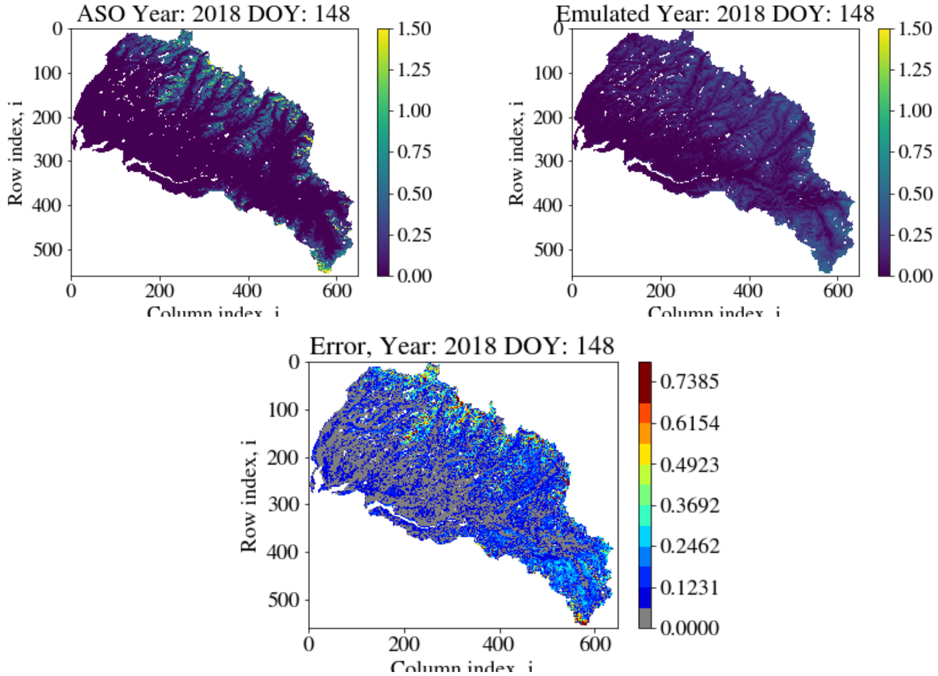

Figure 18: Caption

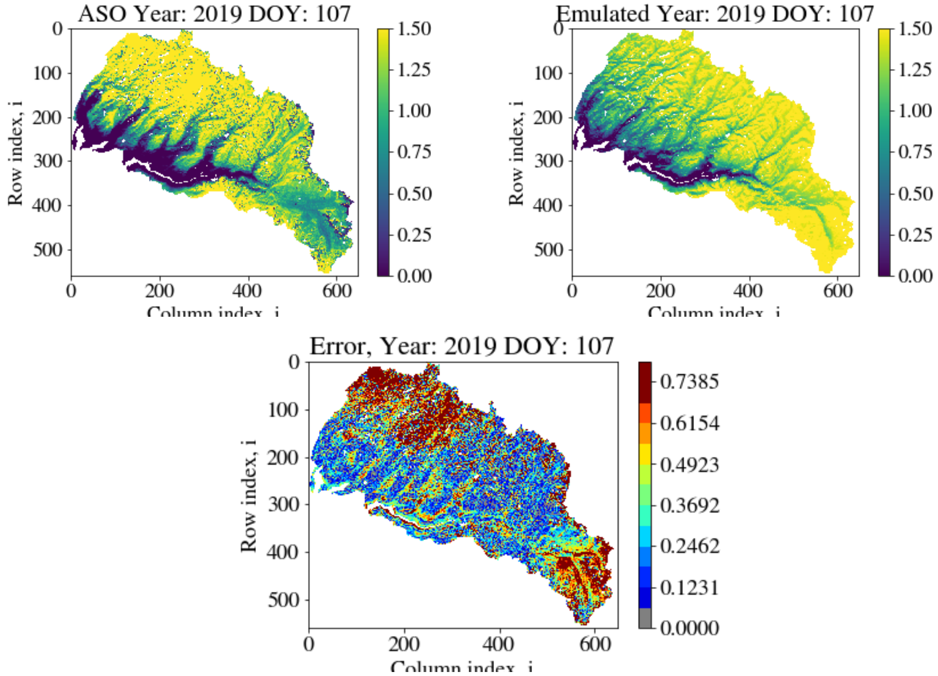

Figure 19: Caption

# 5 Discussion

Of the two models it was much easier to optimise the Random Forest than the MLP, There were clear MSE and $R^2$ curves where one could easily pick out the best value for the hyperparameters: depth and number of trees. For the MLP the situation was quite different, the curves of the hyperparameters: learning rate, epochs and hidden neurons did not give any unambiguous answers as to what was the best value to pick. This was perhaps because i was optimising using the minimum and maximum SWE. In the more extreme events of high and low snow cover it is likely that there are other, more important parameters determining the snow cover than the static terrain parameters i have used. The low $\mathrm{R}^2$ score in figure 7 (228/2017) together with the low snow cover indicates that the static terrain parameters i am using are not sufficient to explain the SWE during this ASO flight. This is likely due to other parameters dominating the snow cover like high air temperature or the winter precipitation being low. If this is the case then the MLP never had a chance to perform well under my investigation. A study modelling SWE using both random forests and MLPs found that both methods performed near similarly well, which indicates that the difference found in my results may be caused by poor optimisation using the MLP, although the study used different predictive variables [6].
The flexibility of the MLP is, from a users point of view, one of its main drawbacks.

There are a near infinite number of hyperparameters that can be optimized: How the network if organised, varying number of neurons per layer, the number of layers, what activation function you use and so on. It is almost impossible to know what the best combination of hyperparameters for a given task is [9]. I could however of used some more automated approaches to find a better combination like grid search with cross-validation or a randomised search [9].

As seen in figure 8 the lowest MSE and the highest MSE occur in the years with respectively the lowest and highest SWE depth. The disadvantage of using the MSE as a test metric in this case is that the magnitude of the MSE will change along with the magnitude of the SWE. A normalised metric would perhaps be better for model evaluation. This said, if we think about how this model could be employed in practice, it is critical that the model performs well in the outlying cases. In a year with a lot of snow in the mountains high melt during the summer may cause floods, burst dams or determine the price of hydroelectric power. If governments or companies who operate water management systems are making decisions based on models with a large error in years with high snow cover, this could lead to catastrophic consequences. Similar studies using random forests and Neural Networks to model SWE have found a larger error in years of anomalously high SWE values, as we see here [7].

It is interesting to see in figure 4 that the bias of the model remains constant, however many trees there are, while the variance decreases. This is however expected since the bias of bagged trees is the same as that of the individual trees. The reduction in error therefore reliant on reducing the variance. The variance of a random forest should decrease asymptotically with a growing number of trees, this means that we can keep increasing the number of trees for as long we want, resulting in a lower and lower variance, given we have the patience to wait long enough for a computation to finish [10]. In practice though there is a limit were the benefit of adding more trees cannot possibly be justified by the growing computational load.

## 6 Conclusions

In this project we saw that the random forest method managed to capture the SWE in the basin rather well using only 20% of the available data for training. There were however issues when it comes to the extreme cases of very high or very low snow cover. The MLP performed poorly and did not accurately capture the distribution of the SWE, though this was likely due to bad optimisation.

With further refinements models like this could reduce the number of flights needed to be taken by the ASO in a given year. You could for instance take measurements on half the basin and use ML to evaluate the rest. This could cut the costs of taking such measurements, and free up time and resources to take measurements in other areas. The severe shortcoming is however that none of the predictors i have used change much with time, they therefore cannot make predictions of future snow cover where there do not exist measurements of the ASO. In a climate that is rapidly warming, static terrain predictors will poorly capture the effects of climate change. Meteorological inputs like air temperature, wind, solar radiation, humidity and so on have been shown to have a large influence on the snow cover [7]. Meteorological data like this is available for many locations globally, and could be incorporated into the model, likely with better results.

# References

[1] sklearn metrics mean squared error, 2020. URL https://scikit-learn.org/stable/modules/generated/sklearn.metrics.mean_squared_error.html.

[2] sklearn metrics r2 score, 2020. URL https://scikit-learn.org/stable/modules/generated/sklearn.metrics.r2_score.html.

[3] sklearn random forest, 2020. URL https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html.

[4] sklearn decision trees, 2020. URL https://scikit-learn.org/stable/modules/tree.html#tree-algorithms-id3-c4-5-c5-0-and-cart.

[5] sklearn multi-layer perceptron, 2020. URL https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPRegressor.html#sklearn.neural_network.MLPRegressor.

[6] E. H. Bair, A. Abreu Calfa, K. Rittger, and J. Dozier. Using machine learning for real-time estimates of snow water equivalent in the watersheds of afghanistan. The Cryosphere, 12(5):1579–1594, 2018. doi: 10.5194/tc-12-1579-2018. URL https://tc.copernicus.org/articles/12/1579/2018/.

[7] David Buckingham, Christian Skalka, and Josh Bongard. Inductive machine learning for improved estimation of catchment-scale snow water equivalent. Journal of Hydrology, 524:311–325, 2015. doi: 10.1016/j.jhydrol.2015.02.042.

[8] Tom G. Farr, Paul A. Rosen, Edward Caro, Robert Crippen, Riley Duren, Scott Hensley, Michael Kobrick, Mimi Paller, Ernesto Rodriguez, and Ladislav et al. Roth. The shuttle radar topography mission. Reviews of Geophysics, 45 (2), 2007. doi: 10.1029/2005rg000183.

[9] Aurélien Géron. Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow. OReilly Media, Inc., 1 edition, 2017.

[10] Trevor Hastie, Jerome Friedman, and Robert Tisbshirani. The Elements of statistical learning. Springer, 2017.

[11] Morten Hjorth-Jensen. Lecture notes on machine learning, fys-stk3155/4155. 2020. URL https://compphysics.github.io/MachineLearning/doc/pub/week44/html/week44.html.

[12] Morten Hjorth-Jensen. Week 45: Random forests and boosting, fys-stk3155/4155. 2020. URL https://compphysics.github.io/MachineLearning/doc/pub/week45/html/week45.html.

[13] Thomas H. Painter, Daniel F. Berisford, Joseph W. Boardman, Kathryn J. Bormann, Jeffrey S. Deems, Frank Gehrke, Andrew Hedrick, Michael Joyce, Ross Laidlaw, and Danny et al. Marks. The airborne snow observatory: Fusion of scanning lidar, imaging spectrometer, and physically-based modeling for mapping snow water equivalent and snow albedo. Remote Sensing of Environment, 184:139–152, 2016. doi: 10.1016/j.rse.2016.06.018.

[14] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12:2825–2830, 2011.

[15] Dominik Schneider, Noah Molototch, and Jeffrey Deems. Estimating relationships between snow water equivalent, snow 1 covered area, and topography to extend the airborne snow 2 observatory dataset. doi: https://doi.org/10.5194/tc-2017-167.

[16] NASA's SVS. Svs: Tracking snow water equivalent in the tuolumne basin, 2020. URL https://svs.gsfc.nasa.gov/4690.

[17] Limin Yang, Suming Jin, Patrick Danielson, Collin Homer, Leila Gass, Stacie M. Bender, Adam Case, Catherine Costello, Jon Dewitz, and Joyce et al. Fry. A new generation of the united states national land cover database: Requirements, research priorities, design, and implementation strategies. ISPRS Journal of Photogrammetry and Remote Sensing, 146:108–123, 2018. doi: 10.1016/j.isprsjprs.2018.09.006.