Lab 4: Cloud Data - Stat 215A, Fall 2017

Briton Park
Hector Roux de Bézieux

November 16, 2017

Abstract

This laboratory project aims to detect clouds over the arctic regions. The MISR sensor aboard the NASA satellite Terra captures radiance levels from locations and transform them into images. Using those radiance levels, this project develops several classifiers to predict the presence of clouds at the pixel level. Those classifiers are run using cross-validation and evaluated based on AUC. The best classification technique is then chosen and the evaluated to assess performance on future new images. We find that the feed forward neural net is the most accurate and most stable classifier

1 Introduction

As global warming becomes more and more a reality, specific climate watch on the arctic region becomes of crucial importance. Indeed, the poles are one of the regions where global warming has the most impact. In turn, retreating sea ice also accelerate global warming. As ground observations are challenging due to difficulties in access and extreme temperatures, being able to collect climate data from satellites would greatly help to study cloud coverage, both locally and across the whole region.

The aim of this study is therefore to use satellite images and develop a proper classifier to detect the presence of clouds, pixel per pixel. To obtain such result, after proper exploratory data analysis, we train several classifiers, compare them on appropriate metrics (see below) and select the best method for further analysis. For this lab project, we used a range of techniques that will be described in greater detail below: k-nearest neighbors, logistic regression, random forest and feed-forward neural net.

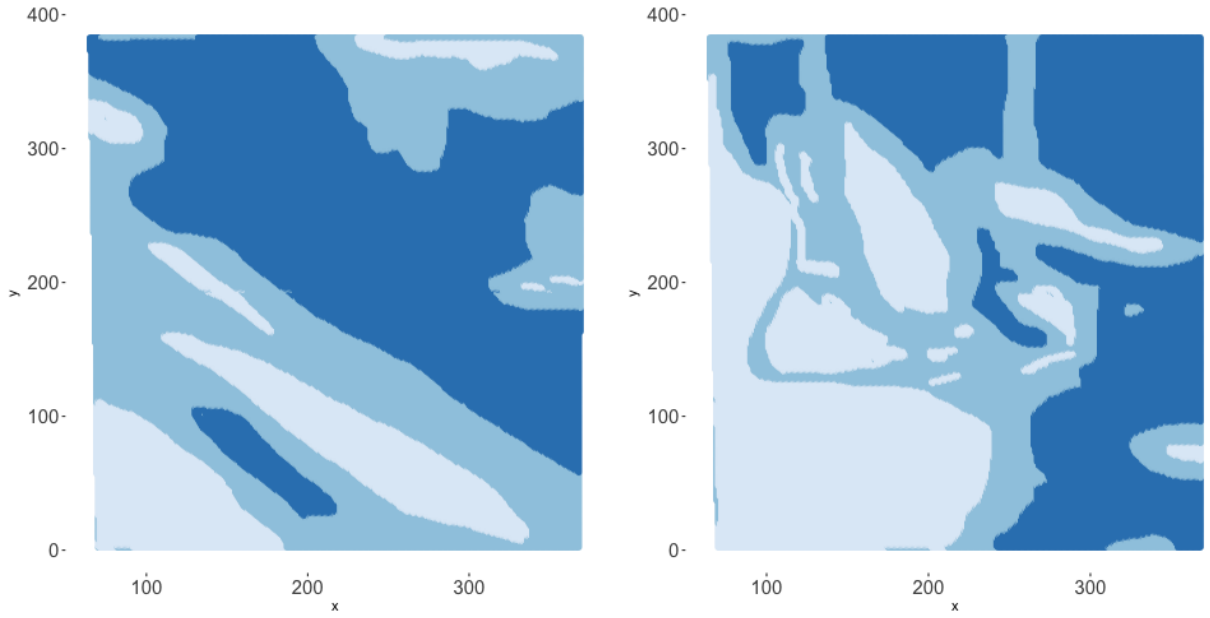
2 EDA

2.1 The data

The data at disposal for training our classifiers consists of 3 images, seen in Figure 1. For each image, we have expert label for every pixel, with the label "high certitude cloud", "high certitude ground / ice" and "unknown". All pixels also come with 8 measurements of intensity: NDAI, SD, CORR, DF, CF, BF, AF and AN. The first three features come from the Multiangle Imaging SpectroRadiomter (MISR). NDAI is a normalized difference angular index that characterizes the changes in a scene with changes in the MISR view direction, CORR is the correlation of MISR images of the same scene from different MISR viewing directions, and SD is the standard deviation of MISR nadir camera pixel values across scenes (Yu 2008). DF, CF, BF, AF, and AN are radiances of different angles obtained from the MISR.

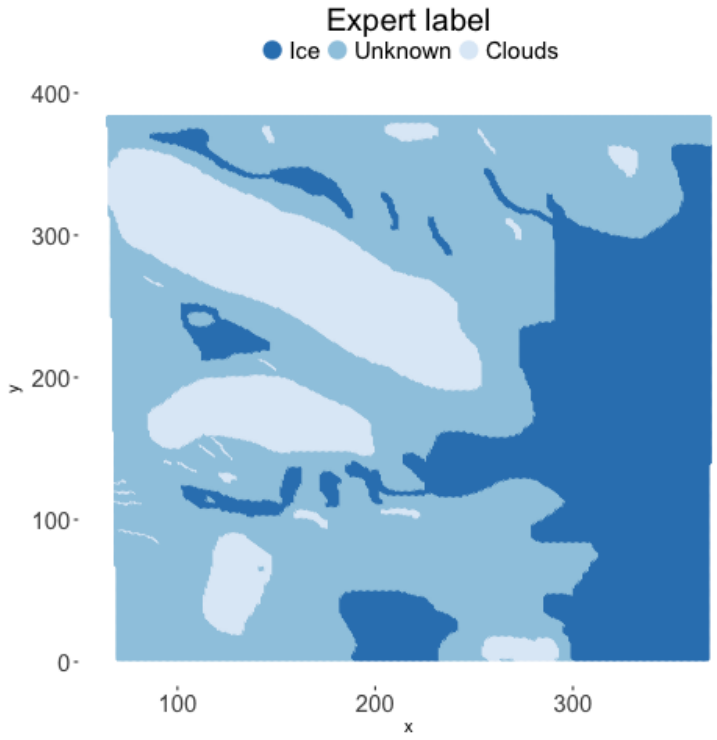
2.2 Relationships between measurements

We visually explored the relationships between radiances of different angles between cloud and non-cloud points. Due to constraints on the report size, we only include scatterplots of the relationship between the



(a) Expert label on Image 1

(b) Expert label on Image 2



(c) Expert label on Image 3

Figure 1: Expert labels of all images

radiances DF and CF, BF and CF, and AN and AF (Figure 2). We found that for both cloud and non-cloud points, there were strong positive correlations between the different angle radiances. However, the relationships were tighter for non-cloud points even though there are less cloud points (80,981) than non-cloud points (127,080). Additionally, in the scatterplots of the first two sub figures (Figure 2), there seems to be two overlapping groups of points for the cloud points. This structure in the data is not present in the non-cloud points.

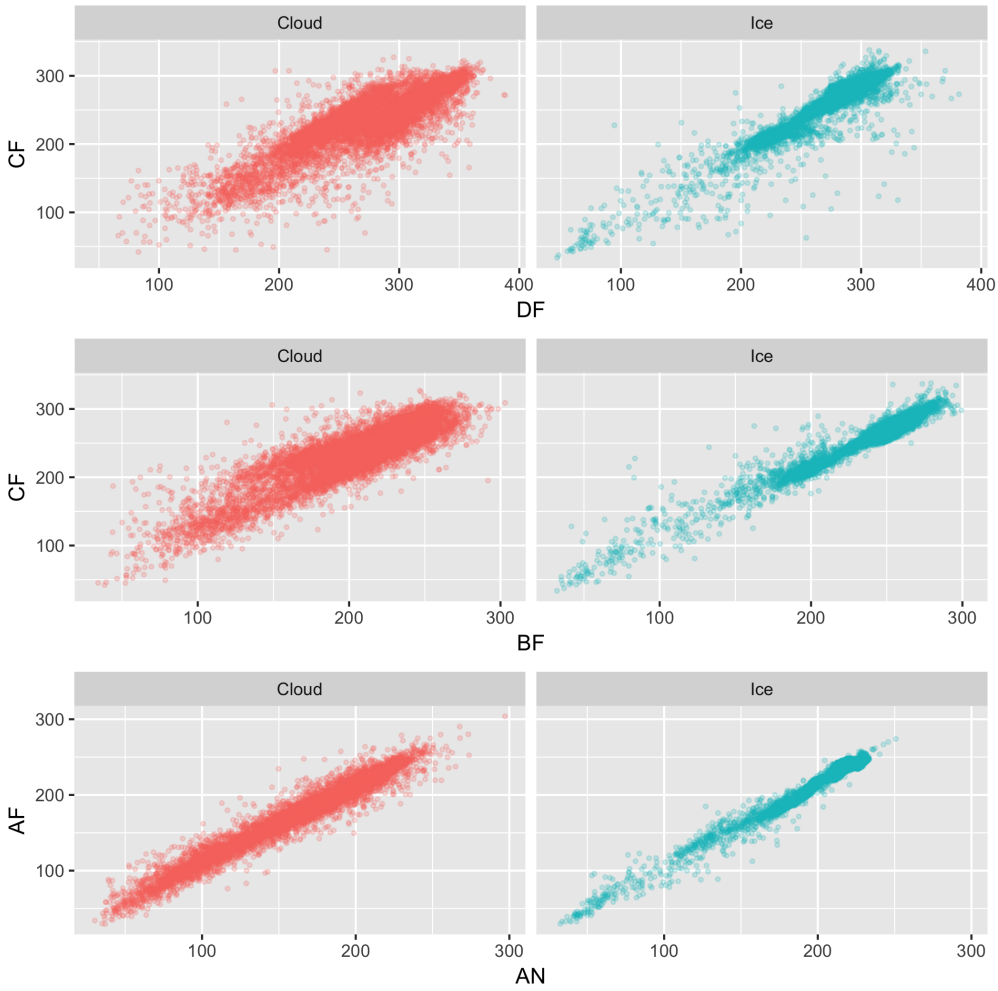


Figure 2: Scatterplot of different radiance measurements

We also quantitatively assessed the relationships between the different radiances for the cloud and non-cloud points. For the non-cloud points, the average correlation between all pairwise radiances was 0.936, while the average correlation between all pairwise radiances for the cloud points was 0.803. These findings agree with our visual observation of the radiance relationships. Because there are strong positive relationships between the radiance angles for both cloud and non-cloud points, there are redundancies in the data that can be removed for modeling purposes.

We then assessed the NDAI, CORR, and SD variables using the similar methods. From the scatterplots (Figure 3), we can see that there are slightly positive relationships between the variables for both the non-cloud and cloud points. The relationships for the cloud points were much weaker than those of the non-cloud points. For the relationships between NDAI with CORR and SD (Figure 3), we can see that the variance in the relationship becomes greater as the NDAI variable value becomes greater than two for the non-cloud points. The pairwise correlations between the variables for cloud points was 0.53 and for non-cloud points was 0.621. These values agree with our visual findings of the relationships between these variables.

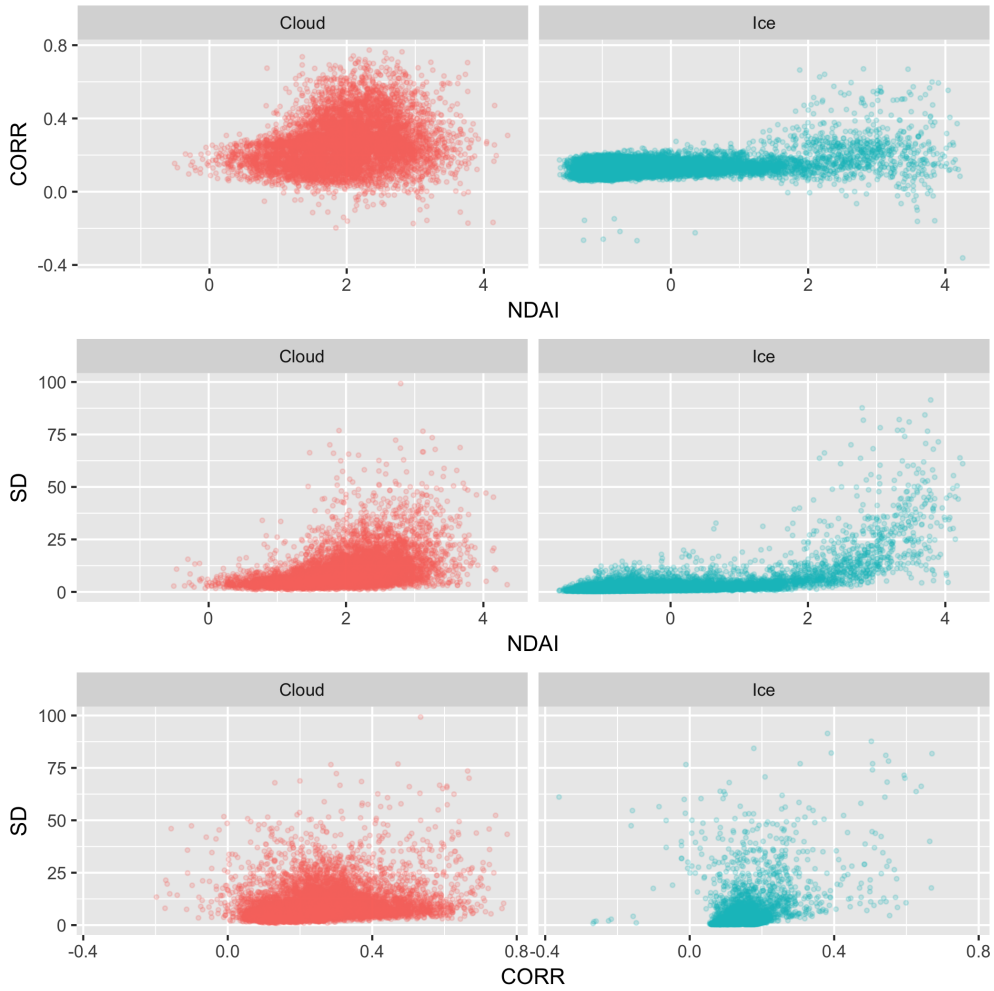


Figure 3: Scatterplot of NDAI, CORR and SD versus one another

3 Modelling

3.1 Reducing the number of features

To assess whether these variables would be helpful in classifying cloud vs non-cloud for the points, we created boxplots using NDAI, CORR, and SD (Figure 4). The variables seem promising for classification, because they had significantly higher values for the cloud points than the non-cloud points. The NDAI variable seems especially useful due to the difference of the 25th, 50th, and 75th percentile values between the cloud and non-cloud points with respect to the range of the values of NDAI. To assess whether these variables are more useful than the radiance values for classification purposes, we will need to use more quantitative methods.

To find the best 3 predictors out of the radiances, NDAI, CORR, and SD, we used forward step-wise regression. This method runs regression models starting with the null model. The model then tests the addition of each variable against the Akaike Information Criterion (AIC). The variable that leads to the lowest AIC is chosen. This process is iterated until the model fit stops improving. The AIC is an appropriate criterion for model selection since it related to the likelihood of the model with an added penalty term on the number of model predictors. The model with the lowest AIC is preferred.

We ran forward step-wise regression and found that the model with only 3 predictors included NDAI, CORR, and SD variables. Note that this does not mean that the best model with 3 predictors is this particular

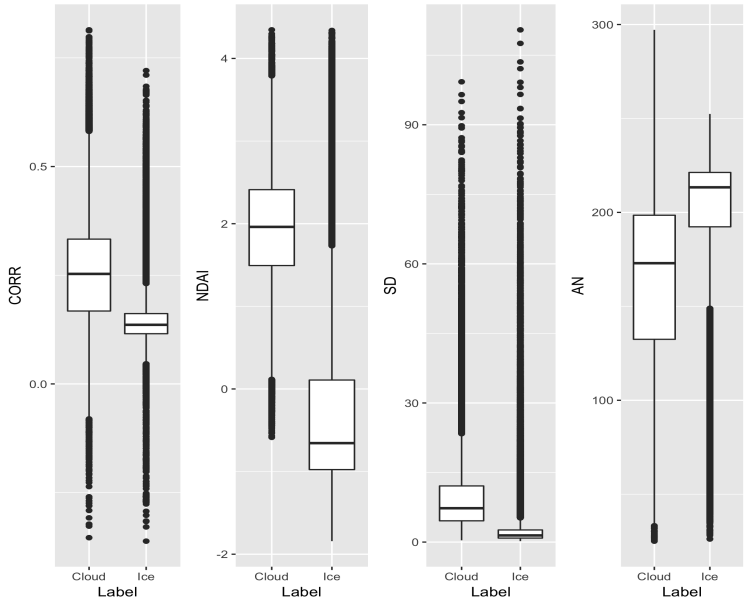


Figure 4: Boxplots for NDAI, CORR, SD and AN for cloud versus non-cloud points

model, because forward step-wise regression is a greedy algorithm. However, it is an approximation and is more efficient than comparing all possible models with 3 predictors. Thus, the results imply that the best 3 predictors for classifying cloud vs non-cloud points do not include the radiance angles.

3.2 Adding neighbors information

To improve the accuracy of our model, we can also take into account the nearest physical pixels. The assumption is that a pixel next to a pixel labelled "cloud" is much more likely to be labelled "cloud" as well, and likewise non "non-cloud" labels. This assumption is based on the observations on Figure 1: clouds never occupy only a few pixels.

Therefore, we look at the 4 nearest neighbors (using euclidean distance on x and y). Expect on the sides of the images, those points will be those directly on top, on the left, the right and the bottom of the considered pixel. We therefore add the measurements of all four neighbors as information to the pixels. As we will see below, our cross-validation method ensures that this will not lead to upper bias.

We therefore trained all the classifiers both with and without neighbors information.

3.3 Description of classifiers

We developed four classification models to test the presence of clouds: logistic regression, neural networks, random forest, and k-nearest neighbors. We included either all the variables or the 3 selected above in the modeling to be able to compare. Using all features was consistently worse (expect from random forest where it did not matter). Therefore, we will not report those results hereafter.

- **Logistic regression** assumes that there is no multicollinearity in the data. It is clear that there is multicollinearity in the data, especially in regard to the different radiance angles. However, we chose not to fix this issue, because we are interested primarily in prediction and not inference in this report. Thus, we will avoid inferring how the individual predictors affect the response from our model. This method also assumes that the error terms and observations are independent. However, we can see from

the raw images that this assumption is also broken, because points that are neighbors are more likely to have the same label. Thus, we should not use the logistic regression model for interpretability and use it for only prediction purposes.

- **K- nearest neighbors** assumes that the closest two points are using euclidean distances on the measurements, the more likely they are to be of the same type (cloud or not cloud). This assumption can be retrospectively checked after running the KNN algorithm. If we get a low error rate, this validate the assumption. We ran the algorithm for k ranging from 3 to 7 and did not notice much difference so we only report the results for $k = 5$.
- **Random forest** is even less parametric but also ignores the multicollineairy of the data. Once again, those assumptions can be verified by the accuracy of the results. We also notice that a forest of 60 trees is enough to reach the asymptotic error rate.
- **Neural networks** on the other hand, do not place any assumptions on the data, error, or response. Thus, there is nothing to test for this method.

3.4 Cross-validation and selection of the best model

To test our methods, we chose to use something similar to cross validation in order to evaluate the methods. To choose the best method out of the ones we have applied, we cut up each of the images into 9 blocks and created 9 sets. Each testing split contained 2 out of the 9 blocks from each image and the training split contained the remaining 7 blocks. The blocks were randomly chosen to be in the test and train splits.

Validation was done this way, because if we randomly partitioned the points in the images into training and test splits, it could be the case that a neighbor of a point in the testing split was in the training split. Thus, the methods would do better than is expected when testing on an entirely new image. Splitting the images into blocks in this way would alleviate some of this problem. We only did 9 out of all 36 possible distinct pairs for computational reasons, as some models are quite long to run. The code to reproduce those sets is available in the folder **R/CV/**

To evaluate our models, we chose to use the AUC metric of the ROC curve. The AUC is a metric where a perfect model would have a value of 1, and a randomly guessing model will have a value of 0.5. It is calculated using the ROC curve which visualizes all possible classification thresholds and determines the discrimination ability of the model by assessing its sensitivity and specificity.

4 Selecting the best model

4.1 Note : Reproducing the results

The code to recompile all results are present in the **R/classifiers/** folder. However, due to the length of computation, they are not reproduced here. Only logistic regression can be run easily on a local machine. The results are stored in the *result.txt* file and contain the AUC for all methods and parameters.

4.2 results

Model	Type	AUC
Logistic	Selected features	0.90
KNN	Selected features Neighbors	0.91
Logistic	Selected features Neighbors	0.93
FFN	Selected features	0.93
KNN	Selected features	0.94
Random Forest	Selected features	0.95
Random Forest	Selected features Neighbors	0.96
FFN	Selected features Neighbors	0.96

There are a few observations to make from the table.

- The results are all above 0.89 for every classifiers so we always do quite better than random guess. The best classifiers score at 0.96, which mean we have high accuracy for those classifiers.
- The addition of the features of the neighbor points really change the results. Only for random forest do we have very consistent results. The biggest difference is for KNN.
- Finally, the less assumption we put on the model, the bigger the AUC. As we could have expected, the models under their respective optimal choices of parameters always perform in the following order (from worst to best): logistic, KNN, random forest and neural net.

However, the first few classifiers are really close and the mean AUC is not enough to allow us to pick the best. Looking at the variance to see which is the most stable might be interesting but the variance of the two best models differ by only 0.7%. Therefore, we look at the boxplot of AUC scores across all tested CV sets, in Figure 6.

As we can see, we have a wide variation depending on the CV set being considered. We can also see that the two set where the classifier performs the worst are always set 4 and 5. When we look at those set, we can see that they both contain square 4 of the images, which is the top middle square. It is probable this location is really hard to predict for at least one of the images.

Because the neural net clearly outperform the random forest on those two sets, **we pick the neural net as our best classifier.**

5 Result on the best model

5.1 Cross-validation on the best model

We chose to use cross validation with respect to the image to assess how well our best model does. Three cross validation sets were created by leaving out image 1, image 2, and image 3 for sets 1, 2, and 3, respectively. This was done because we want to see how well our final model would perform on an entirely new image.

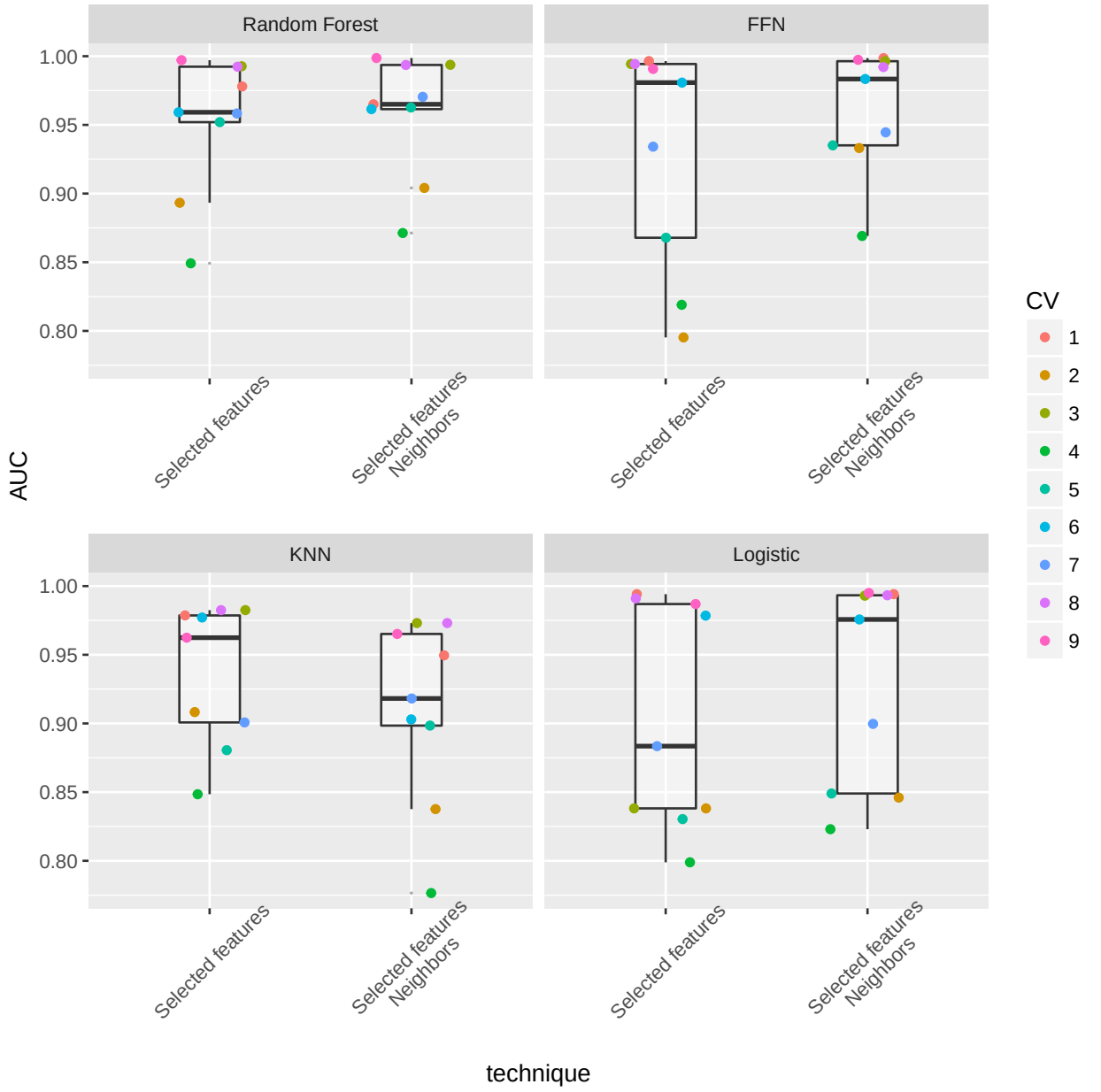


Figure 5: AUC results for all classifiers

5.2 Best model details

As mentioned above, the neural network using the features of the neighboring points is the best model. The neural network model is a feed forward neural network with 5 sets of inputs, which were the feature points for the 5 points (original + 4 neighbors) used for each observation (See supplementary information for more information). This model has an input layer for each of the feature points, an additional dense layer with 250 hidden nodes that takes input from the 5 input layers and an output player of 125 nodes.

To assess the convergence of the algorithm, we used a remote user-interface server from deeplearning4j's functionality which obtains details of the training process. From the interface, we obtained a plot of the training iteration number against the model scores using the first cross validation set of the first two images

as an example (Figure 6). We note that the scores are somewhat noisy in the first 30,000 iterations and become more stable past 35,000 iterations which gives evidence that the model converges.

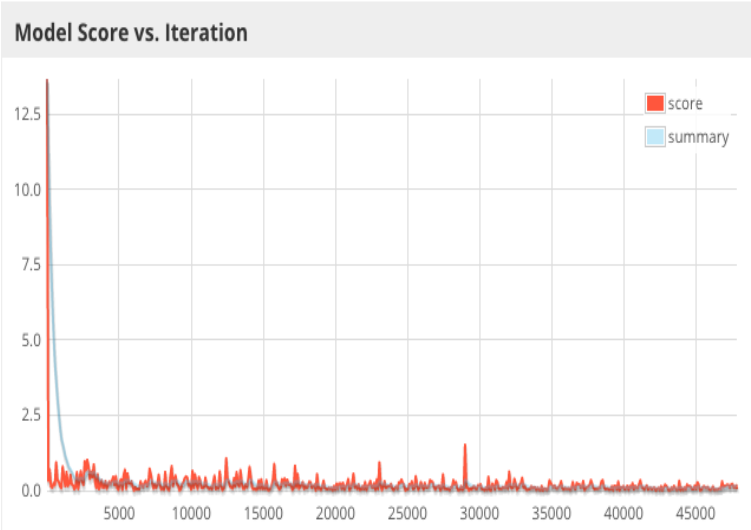


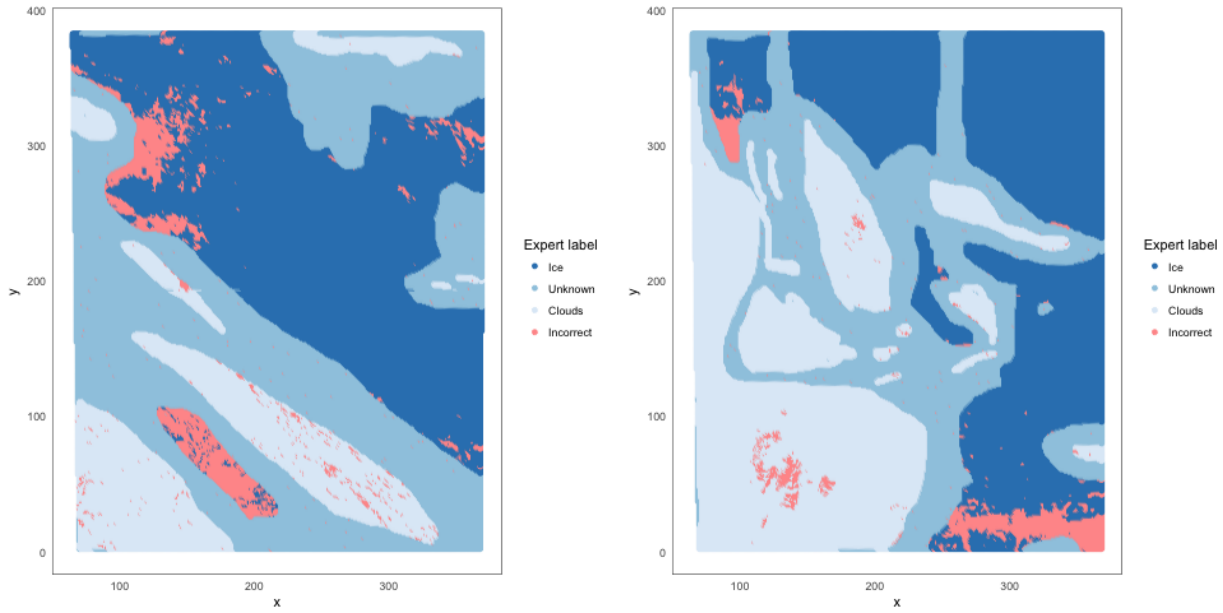
Figure 6: Iterations vs model scores

We then studied the misclassification errors of the neural network. To do this we plotted the misclassified points along with the correctly classified points for each of the images by training on the other two images (Figure 7). To classify the points, we found the best threshold value to convert probabilities into labels. From the images, we can see that there are specific places where the neural network fails to classify the points well. For example, we can see that most of the points in the region of ice in the bottom of image 1 are misclassified. In image 3, most of the isolated regions of ice are misclassified. Furthermore, the neural network seems to do much better at classifying points with the cloud label correctly than for points with ice label.

We also predicted the unlabeled points for each of the images by training on the other two images (Figure 8). We can see that for the most part most points are classified similarly as their neighbors and that there are distinct regions of ice and snow in image 2. In images 1 and 3 there are more regions where many points are classified as ice and cloud. Thus, it seems that the model does the best on image 2, since we expect points of ice and cloud to comprise continuous regions of ice and cloud.

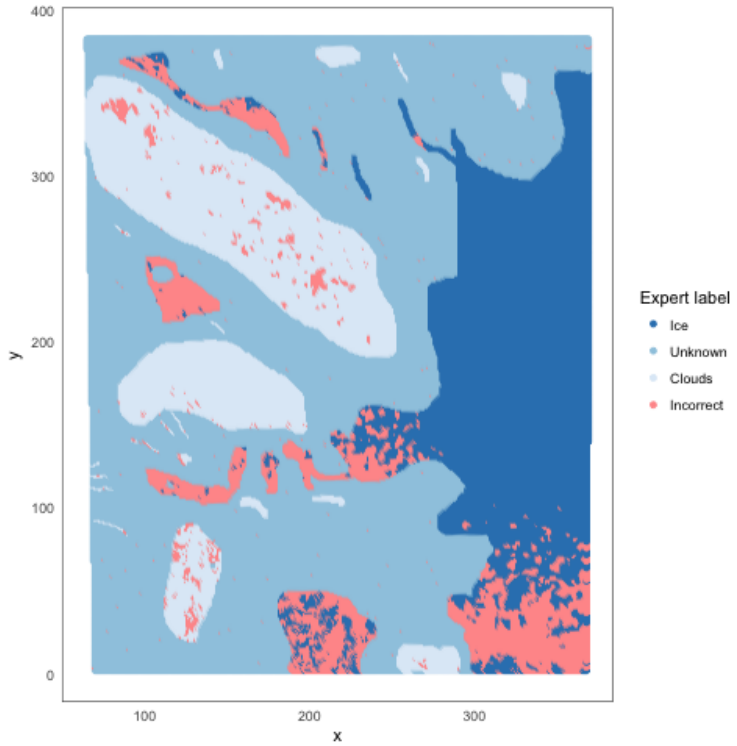
To quantitatively assess the missclassification, we calculated the percentage of misclassified examples for cloud and non-cloud points. For image 1, 2, and 3, the percentages of errors for cloud points were 3, 2, and 5 percent respectively. The corresponding percentage of errors for non-cloud points were 11, 7, 26 percent, respectively. So, as can be seen in the images, we do better at predicting cloud label than non-cloud label.

We then computed the percentage of error for points that had at least one unlabeled neighbor but were not unlabeled themselves (i.e "border" points). For images 1, 2, and 3, these were 7, 17, and 33 percent, respectively. We then compared these percentages with the percentage of error for points that had all labeled neighbors. For images 1,2, and 3 these were 0.9, 5, and 17 percent respectively. Thus, we see that for images 2 and 3, the error percentages were higher than the case where at least one neighbor is unlabeled. For image 1, these errors were very close. Thus, it gives some evidence that "border" points are harder to classify, as can be expected from Fig 8.



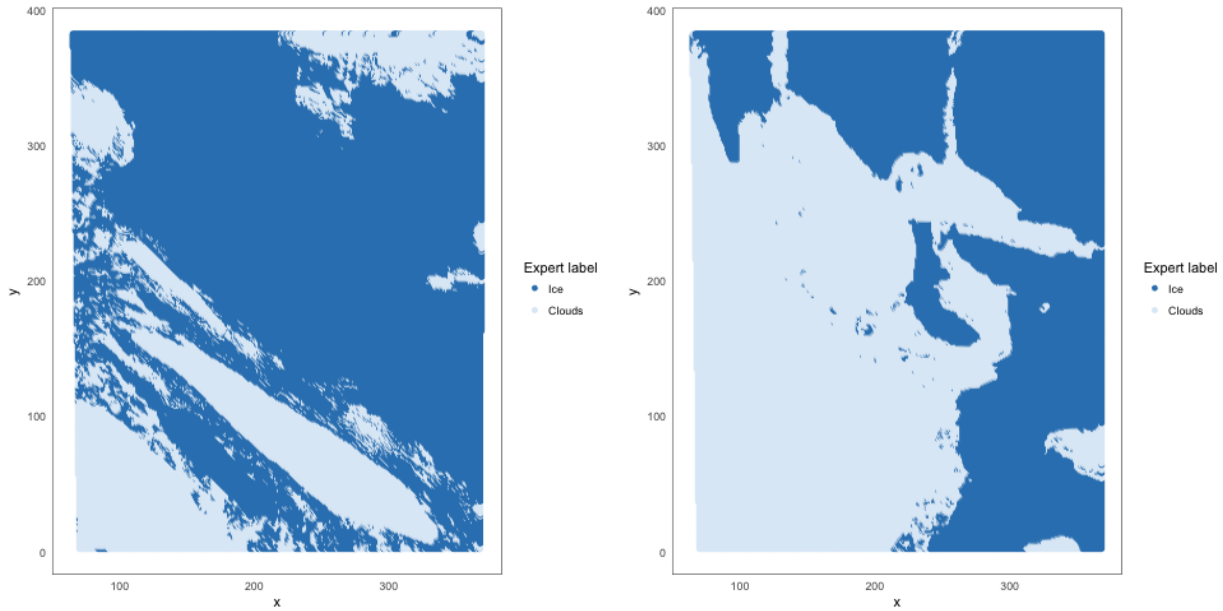
(a) Expert label on Image 1

(b) Expert label on Image 2



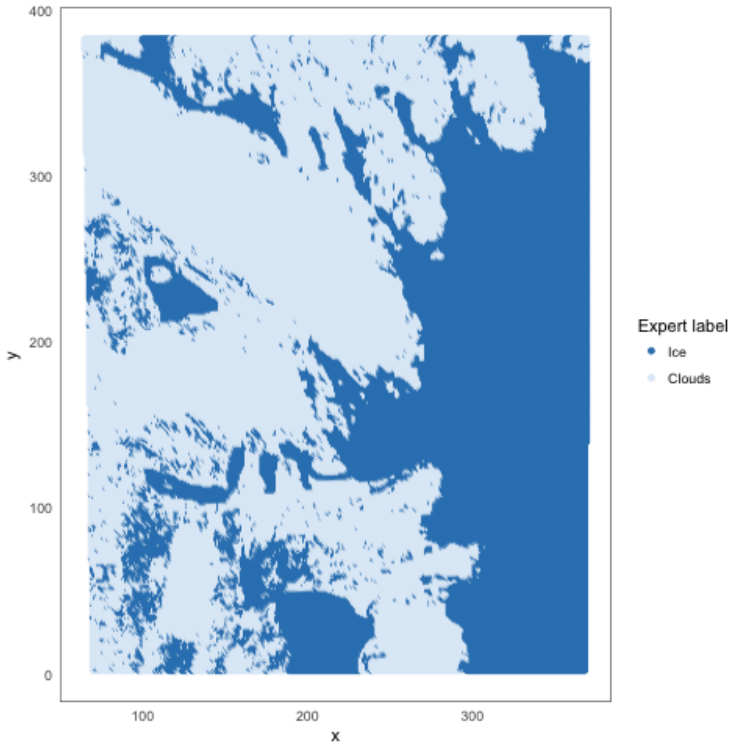
(c) Expert label on Image 3

Figure 7: Misclassified labels on images



(a) Expert label on Image 1

(b) Expert label on Image 2



(c) Expert label on Image 3

Figure 8: Assessing labels for unlabeled points

6 Conclusion

With our analyses, we believe our model would work well on future data. We tested our model using entire images that it has not been trained on and have confirmed that it generalizes well. In addition, because we only train the model on at most 2 images, we believe it will perform better on future data since we would use the full available data (3 images) to train the model. Furthermore, we would expect our model to perform better on points with cloud labels. Future work could test whether different submodels of the feature set would perform better and whether averaging predicted labels of the neighboring points as well as the original would serve as a good predictor.