Appears in Journal of Aerospace Information Systems (JAIS), 2024. https://doi.org/10.2514/1.I011318

# Storm Classification and Dynamic Targeting for a SMart ICE Cloud Sensing (SMICES) Satellite

Jason Swope *, Steve Chien †, Xavier Bosch-Lluis‡, Emily Dunkel§, Qing Yue ¶, Mehmet Ogut‖, Isaac Ramos**, and Pekka Kangaslahti††
*Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA 91109, USA*‡‡

William Deal§§ and Caitlyn Cooke¶¶
*Northrop Grumman, Redondo Beach, CA, 22042*

**Smart Ice Cloud Sensing (SMICES) is a small-sat concept in which a radar intelligently targets ice-storms based on information collected by a lookahead radiometer. Often space observations are performed by continuously collecting data from an instrument aimed at nadir (e.g. directly below the space platform). However, if the platform has the ability to assess science utility of features that will be overflown, an intelligent measurement scheme can improve science return. In the case of SMICES, power constraints and the rarity of storms means that with blind nadir targeting SMICES would collect a limited amount of ice storm radar data. The work proposed acquires measurements to maximize acquired high interest storms while concurrently collecting a background sampling of all features. This is accomplished through two steps: storm classification and dynamic targeting. For classification we describe multi-step use of Machine Learning and Digital Twin of Earth's atmosphere to create a classifier. We discuss an autonomous data labelling pipeline used to train five different models to identify storms in tropical and non-tropical regions and assess the results and impact of expected noise. For dynamic targeting six algorithms ranging from "blind" to more selective are described and evaluated on their improvement over "blind" targeting.**

*Member of Technical Staff, Jet Propulsion Laboratory, California Institute of Technology, 4800 Oak Grove Drive, Pasadena, CA 91109, USA
†Engineering Fellow, Senior Research Scientist, and Group Supervisor, Jet Propulsion Laboratory, California Institute of Technology, 4800 Oak Grove Dr, Pasadena, CA 91109, USA
‡Senior Member of Technical Staff, Jet Propulsion Laboratory, California Institute of Technology, 4800 Oak Grove Drive, Pasadena, CA 91109, USA
§Senior Member of Technical Staff, Jet Propulsion Laboratory, California Institute of Technology, 4800 Oak Grove Drive, Pasadena, CA 91109, USA
¶Member of Technical Staff, Jet Propulsion Laboratory, California Institute of Technology, 4800 Oak Grove Drive, Pasadena, CA 91109, USA
‖Member of Technical Staff, Jet Propulsion Laboratory, California Institute of Technology, 4800 Oak Grove Drive, Pasadena, CA 91109, USA
**Member of Technical Staff, Jet Propulsion Laboratory, California Institute of Technology, 4800 Oak Grove Drive, Pasadena, CA 91109, USA
††Senior Member of Technical Staff, Jet Propulsion Laboratory, California Institute of Technology, 4800 Oak Grove Drive, Pasadena, CA 91109

§§Departmental Staff Engineer, Northrop Grumman, Falls Church, VA
¶¶RF/Microwave Design Engineer, Northrop Grumman, Redondo Beach, CA

# I. Introduction

High altitude ice clouds, covering more than 50% of the Earth's surface, are often produced from high-impact deep convection events [1], and are strong modulators of Earth's weather and climate [2, 3]. High altitude ice clouds play a significant role in the Earth's energy balance and hydrologic cycle through their effects on radiative feedback and precipitation, and therefore crucial for life on Earth.

Smart Ice Cloud Sensing (SMICES) is a small-sat concept designed to increase our knowledge of the phenomena by collecting data on the vertical resolution of the ice cloud particles. This information has never been analyzed through a global satellite. Instead, only in-situ ice particle size data has been available. Therefore SMICES will be able to provide an innovative path towards the quantification of how ice cloud radiative effects impact convective storm intensity, size, and track as well as constraining climate model simulations of ice cloud feedbacks and associated hydrological processes, contributing to reducing uncertainties of climate predictions.

This paper focuses on the classification and targeting systems of the SMICES mission. We describe the use of unsupervised machine learning, digital twin (Global Weather Research and Forecasting (GWRF) model), and supervised machine learning to address the challenge of developing a radiometer-based classifier for deep convective ice storms. The knowledge gained from this classification is then used to improve the targeting system of SMICES. Current targeting systems for satellite imaging consist of continuously targeting nadir. If the instrument cannot be on for the entire duration of the orbit, it is randomly turned off to meet the energy restraints. This method ensures that the data collected will be either the ground-track of the satellite, or a random subsample of that. It also only takes images at nadir, which is the angle that returns the best data. This approach fails to address the problem that some parts of the sky are more intriguing than others. Smart targeting can be used to guide an instrument to focus its analysis on the more interesting areas of the sky as the instrument flies over.

The goal of SMICES is to analyze clouds, specifically deep convective storms. Deep convective storms are extremely rare even though global cloud coverage can span roughly $2/3$ of Earth [4]. While SMICES would be able to collect data on some storms using general image targeting techniques, its performance can be improved with development in the pathing of its scientific instrument. SMICES intends on utilizing the knowledge gained from its cloud classification to allow its planning algorithms to target the most scientifically interesting clouds along its path. Additionally, because the radar uses a lot of energy, SMICES will only be able to capture data for about 20% of the time. These constraints lead to the need for active targeting, which will allow the mission to maximize its scientific return by selecting which clouds are analyzed.

The remainder of this paper is organized as follows. In Section 2 we review related work. In Section 3 we provide context on the classification problem and an overview of the classifiers and data that will be used. Section 4 describes the tropical and non-tropical datasets and an automatic labelling system we establish. Section 5 and 6 explain the different classifiers utilized and how they were set up for this experiment. Section 7 reviews the results of each classifier

on both datasets. Section 8 explores the impact of noise on the accuracy of the classifiers. Section 9 provides context on the smart targeting problem. Section 10 and 11 introduce the algorithms developed and explain how they were set up for the experiment. Section 12 reviews the results of the algorithms. Section 13 discusses the improvement smart targeting provides when classification errors are taken into consideration. Finally, Section 14 and 15 outline future work and summarize our conclusions.

## II. Related Work

### A. Classification

There are many examples in the literature of using supervised learning for cloud classification. Artificial neural networks and support vector machines (SVMs) have been used to classify clouds in RGB imagery from a ground-based camera, using the pixel values as model inputs [5]. In other work, summary features from nearby pixel regions have been used as classifier inputs. This includes textural or statistical features as inputs to a k-nearest neighbor (k-NN) classifier [6] [7] and artificial neural network and SVM classifiers [7]. In addition, convolutional neural networks (CNNs) operating over image patches have been used on RGB imagery [8]. A method named bag of micro-structures (similar to bag-of-words in text), represented the imagery with a weighted histogram of micro-structures and passed this into an SVM classifier [9]. Other work has involved ground-based infrared (IR) imagery, using structure features in combination with the rectangle method, which uses upper and lower significant bounds, for cloud classification [10].

In our work, we use supervised learning for storm classification at the pixel level, focusing on higher altitude clouds. This differs from cloud classification because we are only interested in identifying the small subset of clouds that are storms. Instead of IR or EO imagery from a ground-based camera, we use simulated radiometer data from a satellite. In addition, we use a digital twin in combination with sparse labeling and unsupervised clustering to facilitate label generation.

Prior work has explored the use of unsupervised learning to cluster cloud data. Clustering has been explored using three-dimensional histograms applied to multi-spectral satellite imagery, using the visible, IR, and water-vapor channels [11]. Later work improved upon this by introducing textural parameters and processing larger datasets at different times [12]. Another work attempted to reproduce the class clusters using Probabilistic Self-organizing Maps [13].

Our clustering approach differs from the above work, in that we use simulated science parameters from a digital twin to do clustering, as opposed to using features derived from imagery. To the best of our knowledge, there is currently no other work with digital twins being used for the purpose of facilitating label generation in cloud imagery.

### B. Targeting

We focus on the targeting of clouds to guide the radar through the storms. Prior work has addressed storm targeting over a global dataset through various dynamic targeting algorithms [14][15].

Similar work has also been conducted on the inverse problem of cloud avoidance. Cloud screening onboard aircraft has worked to help cut out swaths of data compromised by cloud cover to reduce the amount of downlinked information [16]. This work has been demonstrated on both AVIRIS and EMIT data at NASA Jet Propulsion Laboratory [17]. While SMICES is trying to collect the most useful data, its algorithms are designed to control the data collection process instead of discard invalid portions of previously collected data.

Other implemented cloud avoidance work has been completed on TANSO-FTS-2 where intelligent targeting is utilized to minimize the amount of cloud coverage captured in its images [18]. Other work that focuses on developing algorithms to achieve cloud avoidance is being done at NASA Jet Propulsion Laboratory where a greedy and a graph search based algorithm has been developed to select the most clear sections of sky during a flyover [19]. Further cloud avoidance work has been implemented over a global dataset to more accurately assess the potential benefits of dynamic targeting [20] This work is more similar to SMICES as its goal is to target its instrument at more scientifically relevant features during flight. However, the algorithms differ in their actual targets. Storms are significantly rarer than clear sky and are composed of different types of clouds. SMICES prioritizes two different types of clouds through its flight in contrast to the single feature of clear sky addressed in cloud avoidance. Previous work has explored the SMICES targeting algorithms without considering the impact of classification error [21].

### C. Onboard Processing

Previous work has demonstrated the efficiency of the various classifiers and targeting algorithms discussed in this paper. They were tested on traditional flight hardware and more powerful processors in various environments including the International Space Station [22][23]. The testing proved that on a RAD750 [24] (onboard processor on the Mars Reconnaissance Orbiter and Mars Perseverance Rover) the algorithms ran fast enough for use in flight.

## III. Classification Background

The SMICES classification problem is to correctly classify storms utilizing the information that the on-board radiometer collects. To achieve this the classifier needs to be able to correctly identify five different cloud types: clear sky, thin cirrus, cirrus, rainy anvil, and convection core. The different cloud types have varying levels of scientific significance, with their rankings demonstrated in Figure 1. The convection core and rainy anvil clouds are both considered storms. With respect to the SMICES targeting algorithms, only the rainy anvil and convection core clouds are actively targeted. Therefore, it is extremely important for the classifiers to be able to distinguish between the rainy anvil, the convection core and the three non-storm cloud types. An accurate classification of these cloud types has been shown to enable a gain of capturing convection core clouds by a factor of 24 and rainy anvil clouds by a factor of 2 [21].

We first use clustering and human experts to label the digital twin / GWRF data. The digital twin datasets include three scientific variables of ice water path (IWP), median particle size ($\mu m$), and median cloud top height (m), as

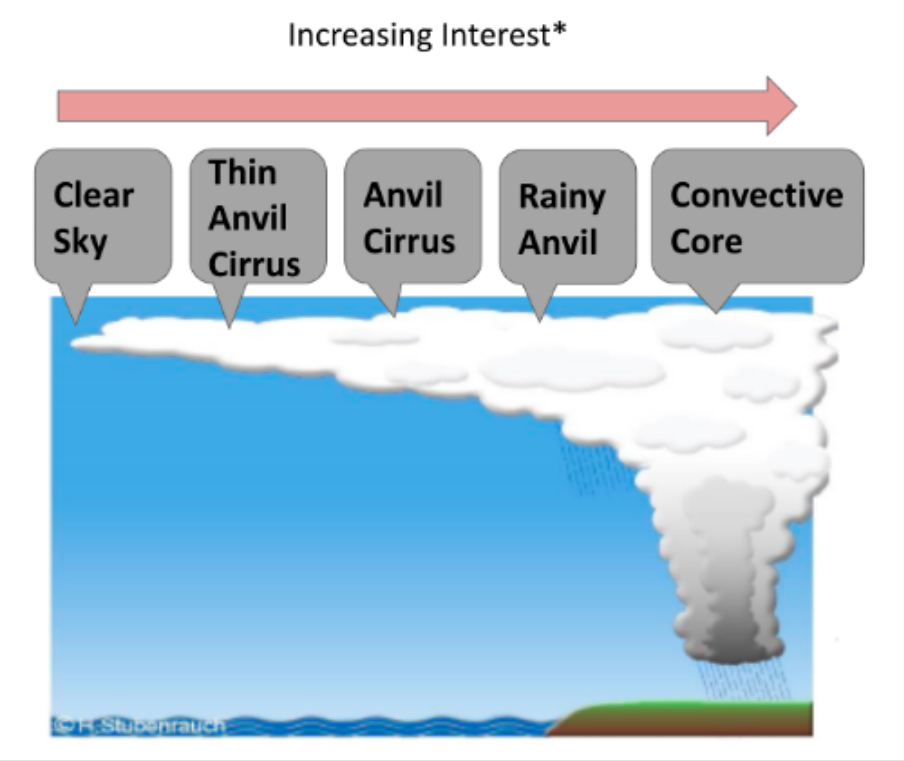


**Fig. 1 Cloud types shown in order of increasing interest**

well as the eight bands of radiance collected by the SMICES radiometer. These three scientific variables are used to automatically generate labels for the data with the help of scientists as explained in Section 4B. This avoids the need to manually label the data.

We then use these labels with the digital twin (simulated) radiometer data to train the classifiers. During testing, the classifiers will only have access to the radiance values since those are the only data available in an operational setting.

A set of classifiers were trained and tested on two separate regional datasets: a tropical dataset over the Caribbean, and a non-tropical dataset of the Atlantic Coast of the United States. The physics of a given cloud type can differ depending on whether it is in a tropical or non-tropical region. For example, there is ice in the deep convective core in non-tropical regions but not in tropical regions, and this affects the radiance values. Thus, we train separate classifiers for each dataset. These classifiers include a random decision forest (RDF), support vector machine (SVM), Gaussian Naïve Bayes, feed forward artificial neural network (ANN), and convolutional neural network (CNN). In an operational setting, we would select the model that corresponds to the geographical region the satellite is over.

# IV. Data

## A. Datasets

Two regional datasets were used in this work: a tropical and non-tropical dataset. Both datasets were created through the Global Weather Research and Forecasting (GWRF) model [25]. The GWRF is a state-of-the-art physics-based weather model. It is used to create computationally expensive datasets that we use as a digital twin to real climate data. In our case the model generated the brightness temperatures for different cloud types along the bands of Tb250+0.0, Tb310+2.5, Tb380-0.8, Tb380-1.8, Tb380-3.3, Tb380-6.2, Tb380-9.5, and Tb670+0.0, as well as the scientific variables of ice water path, median particle size, and median cloud top height.

The tropical dataset is located in the Caribbean. This dataset contains 13 images that are 119x208 pixels with a pixel size of 15km for a spatial extent of 1,785km x 3,120km. Each image is a snapshot of the same area in one-hour intervals.

The non-tropical dataset is located in the Atlantic Coast of the United States. The data contains 29 image cutouts that are 1998x270 pixels with a pixel size of 1.33km for a spatial extent of 2,657km x 359km. The dataset combines to form three images over the same area in 12-hour time intervals. Each larger image is constructed of 10 image cutouts stacked vertically. The total size of a full image is 1998x2700 pixels with a spatial extent of 2,657km x 3,591km. The last cutout of the third time interval was incomplete and therefore left out of this study. Therefore, we only have 29 image cutouts.

The same cloud types in the tropical vs. non-tropical datasets do not necessarily correspond to similar scientific features or radiance values. Because of this, as well as the large difference in resolution (pixel size of 15km vs. 1km), we treat the tropical and non-tropical datasets independently in this work.

## B. Data Labelling

Manually labelling our data was not a feasible option given the extremely large number of pixels, and the difficulty in identifying the cloud type based only on radiance values. We used a digital twin of the radiometer to solve this problem. The digital twin helps in two ways. It gives us access to scientific variables that are not directly measurable in nature and also allows us to generate a large amount of storm data for training and evaluation.

The scientific variables of ice water path, median particle size, and median cloud top height are significant because they give us a way to map a pixel's values to a specific cloud type. Unlike the radiance values, the scientists are able to map these scientific variables to the five cloud types they identified. While this solves the mapping problem, an automated method needed to be developed due to the large number of pixels

The approach to automate the labelling is as follows. First, we cluster the data into representative clusters based on the scientific variables. Then we can map each cluster center to the cloud type that it corresponds to. Each cluster's mapping serves as the label for every pixel within that cluster.

K-means was utilized as our clustering method. This approach requires us to identify the proper number of clusters to accurately represent our dataset before mapping the clusters to the specific cloud types. A failure to accurately

represent our data will cause a poor classification of pixels. The proper number of clusters is determined by analyzing if the number of clusters selected maximizes the within cluster coherence and the between cluster separation. A silhouette score was used to calculate the effectiveness of the number of clusters over the data.

A silhouette score, which ranges from -1 to 1, gives a measure of the cluster's fit. A score of 1 means the clusters are well distinguished, a score of 0 means the clusters are indifferent, and a score of -1 means that the clusters are incorrect. The equation used to obtain this score is: $((b-a)/max(a,b))$ where a represents the intra-cluster distance (average distance between each point within a cluster) and b represents the average nearest cluster distance (average distance between the instances of the next closest cluster). Clustering was performed separately for the tropical and non-tropical datasets.

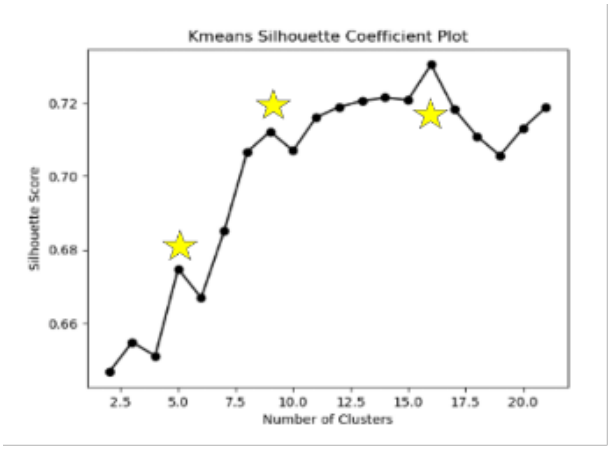


**Fig. 2 Silhouette scores over the tropical dataset**

As shown in Figure 2, there are three local maximums at 5, 9, and 16 clusters in the tropical dataset. It is possible to map multiple clusters from K-means to one cloud type if multiple clusters have similar scientific values. The difference between the silhouette scores of the local maximum at five clusters (.675) is only .06 away from the silhouette score at 16 clusters (.735), which is an insignificant increase. We chose to use five clusters due to the high silhouette score and the easier mapping to the five alluded cloud types.

Silhouette scores were also calculated for a random sample taken from the non-tropical dataset. The scores for all of the clustering values were over .97, which suggests a very good clustering for the dataset. We chose to continue to use a 5 cluster K-means for the non-tropical dataset for its high silhouette score and easy mapping to the original cloud labels.

The scientists assigned each cluster to its respective label based on the mean and standard deviation of its centroid. Overall, the clusters gradually increased in IWP and median particle size while decreasing in median cloud top height. The only exception to this rule was the cluster with all values set to 0, which we assign to the clear sky class. This relationship between the clusters was used by the scientists to identify the labels. In general, a higher IWP and median particle size while having lower median cloud top height correlates to a stronger storm cloud. These mappings for the

Hidden values by cluster

| Cluster Color | IWP (mean: std) | Median Particle Size (mean: std) | Median Cloud Top Height (mean: std) | Assigned Cloud Type |
|---|---|---|---|---|
| Yellow | 0: 0 | 0: 0 | 0: 0 | Clear |
| Orange | .98: 12.9 | 37.2: 16.5 | 11,794: 1,290 | Thin Cirrus |
| Green | 55.2: 197 | 103: 37.9 | 7,412: 787 | Cirrus |
| Red | 473: 806 | 270: 169 | 4,870: 573 | Rainy Anvil |
| Purple | 8,476: 4,312 | 539: 155 | 4,539: 652 | Convection Core |

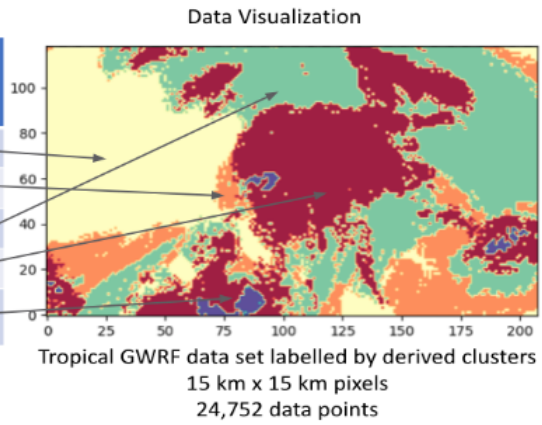


**Fig. 3 Cluster values mapped to their labels on the tropical dataset**

tropical dataset are demonstrated in Figure 3.

## V. Classifiers

The ground truth values for each cloud class are taken to be the labels assigned through the clustering method described above. Now that we have labels, we can train the classifiers to predict the cloud class using only data available in orbit: the radiometer data. During training, the classifiers see both the radiometer data and the class labels, and at test time, see only the radiometer data. The classifiers operate at the single pixel level; given a radiometer measurement at a given pixel, what is the cloud class at that pixel. Thus, our current classifiers do not take into account neighboring pixel values. Classifier performance was evaluated using 5-fold cross-validation, and testing was performed on a separate held-out test set.

For this study we explored the following classifiers: random decision forest (RDF), support vector machine (SVM), Gaussian Naïve Bayesian, a feed forward artificial neural network (ANN), and a convolutional neural network (CNN).

An important factor in the classifier configuration is how balanced the different classes within the datasets are. The classes are heavily skewed away from the most important cloud type, the convection core, in both the tropical and non-tropical datasets.

Table 1 displays the distribution of cloud type in the tropical and non-tropical dataset. The two most important classes, convection core and rainy anvil, only make up around 32% of the dataset. Therefore, if we only classified every pixel as clear sky, thin cirrus, or cirrus we would be able to attain a 68% accuracy, although we would miss the important classes. The class distribution of the non-tropical dataset (Table 1) is significantly more skewed towards clear than the tropical dataset. About 80% of this dataset is non-storm clouds. To help the classifiers overcome these unbalanced datasets, the random forest and support vector machine classifiers are trained with weights adjusted for the class imbalance. When it is noted that the weights are equalized, we are stating that the weights have been rebalanced so

| | Clear Sky | Thin Cirrus | Cirrus | Rainy Anvil | Convection Core |
|---|---|---|---|---|---|
| Full Dataset | 23.8% | 14.0% | 29.2% | 32.0% | 1.0% |
| Train Dataset | 24.1% | 13.6% | 30.0% | 31.3% | 1.0% |
| Test Dataset | 22.9% | 15.7% | 26.3% | 34.2% | 0.9% |

**(a) Distributions over the Tropical Dataset**

| | Clear Sky | Thin Cirrus | Cirrus | Rainy Anvil | Convection Core |
|---|---|---|---|---|---|
| Full Dataset | 60.2% | 10.4% | 9.6% | 15.9% | 3.8% |
| Train Dataset | 59.3% | 8.2% | 11.1% | 17.5% | 3.8% |
| Test Dataset | 62.2% | 15.4% | 6.2% | 12.4% | 3.8% |

**(b) Distributions over the Non-Tropical Dataset**

**Table 1 Cloud type distribution over the tropical and non-tropical datasets and their respective train and test datasets**

that every class is weighted equally.

## VI. Classification Experimental Design

Separate classifiers were trained for tropical and non-tropical regions, using the data corresponding to that region. For the tropical dataset the first ten images were used as a training and validation set. The remaining three images then served as the test set. The test and validation set for the non-tropical dataset was created from the first two time steps. These two timesteps are comprised of the first 20 image cutouts in the dataset. The remaining 9 image cutouts that makeup the final image was used as the test set. The distribution of the cloud classes in each training and test dataset is shown in Table 1. We did not build classifiers using a combined dataset due to the different pixel sizes of each dataset and physical differences found in tropical and non-tropical clouds.

The classifiers are evaluated on their performance by analyzing their accuracy over the three cloud classes of non-storm (clear, thin cirrus, and cirrus), rainy anvil, and convection core. The classifiers were trained in two separate methods:

1) Trained on the original five-class labelling created through K-means. These classifiers output a five-class labelling of the clouds which is converted into the three-class accuracy
2) Trained on three-class labelled data. This data is created by mapping the five-class labels into the three-classes before training. These classifiers output a three-class labelling of the clouds.

The performance of each classifier will be presented over the three-class problem of identifying non-storm clouds (clear, thin cirrus, or cirrus), rainy anvil clouds, and convection core clouds. The simplification from five to three classes is due to SMICES only actively targeting rainy anvil and convection core clouds, with a preference for convection core. Therefore, the meaningful distinctions are between those three classes. However, not all misclassifications are equal. A

| | Clear, Thin Cirrus, and Cirrus | Rainy Anvil | Convection Core |
|---|---|---|---|
| Clear, Thin Cirrus, and Cirrus | 44,226<br>88.38% | 5,751<br>11.49% | 63<br>0.13% |
| Rainy Anvil | 3,858<br>16.55% | 19,043<br>81.70% | 408<br>1.75% |
| Convection Core | 109<br>12.02% | 584<br>64.39% | 214<br>23.59% |

**(a) 5 class labelled data**

| | Clear, Thin Cirrus, and Cirrus | Rainy Anvil | Convection Core |
|---|---|---|---|
| Clear, Thin Cirrus, and Cirrus | 41,339<br>92.39% | 3,361<br>7.51% | 45<br>0.10% |
| Rainy Anvil | 6,816<br>23.83% | 21,365<br>74.68% | 426<br>1.49% |
| Convection Core | 114<br>12.61% | 574<br>63.50% | 216<br>23.89% |

**(b) 3 class labelled data**

**Table 2 Confusion Matrices of the RDF classifier on the tropical dataset, max depth 14, number of trees 32, weights equalized by class**

misclassification of a non-storm cloud as rainy anvil or convection core cloud could trigger the use of radar power for a non-desired target. This would be more costly than a misclassification between a convection core and rainy anvil cloud since the radar would still be collecting scientifically significant data. The performance will also be discussed in the context to the two class problem of non-storm clouds (clear, thin cirrus, and cirrus), and storm clouds (rainy anvil and convection core). This analysis highlights how many significant misclassifications are being made by each classifier.

## VII. Classifier Results

### A. Random Decision Forest (RDF)

We use scikit-learn's implementation of the Random Forest Classifier in this work [26]. We found that 32 trees and a maximum depth of 14 was optimal based on the training and validation sets.

One challenge with the storm datasets is their unbalanced nature. Storms are quite rare in the sky, and the center of storms (convection core clouds) are even rarer. Therefore, the classifier is going to be more prone to classifying clouds as non-storm clouds, since that is the most dominant class. To solve this, the weights have been adjusted in the RDF so that every class is weighted equally.

Table 2 shows the RDF performance when trained on five-class labelled data and three-class labelled data with weights equalized. The most notable difference is the drop in performance in the rainy anvil class and increase in

| | Clear, Thin Cirrus, and Cirrus | Rainy Anvil | Convection Core |
|---|---|---|---|
| Clear, Thin Cirrus, and Cirrus | 3,539,649<br>89.61% | 246,916<br>6.25% | 163,505<br>4.14% |
| Rainy Anvil | 254,077<br>46.25% | 196,388<br>35.75% | 98,939<br>18.01% |
| Convection Core | 273,330<br>76.85% | 38,958<br>10.95% | 43,378<br>12.20% |

**(a) 5 class labelled data**

| | Clear, Thin Cirrus, and Cirrus | Rainy Anvil | Convection Core |
|---|---|---|---|
| Clear, Thin Cirrus, and Cirrus | 3,404,214<br>91.51% | 215,619<br>5.80% | 100,153<br>2.69% |
| Rainy Anvil | 433,653<br>51.12% | 341,693<br>40.28% | 72,887<br>8.59% |
| Convection Core | 228,807<br>79.75% | 44,321<br>15.45% | 13,793<br>4.81% |

**(b) 3 class labelled data**

**Table 3 Confusion Matrix of the RDF classifier on the non-tropical dataset, max depth 14, number of trees 32, weights equalized**

accuracy of the non-storm class for the three-class classifier when compared to the five-class classifier. Overall the three-class classifier returns roughly 7% lower accuracy in the rainy anvil class and an improvement of 4% in the non-storm class. The accuracy of convection core clouds was effectively unchanged. Despite the poor classification of convection core clouds, the classifiers were able to distinguish roughly 76% and 83% of the storm clouds correctly in the three-class and five-class classifiers respectively. Another metric to factor in is the precision for storm clouds. This metric is calculated by dividing the total number of pixels correctly classified as storm clouds (true positives) by the total number pixels classified as storm clouds (true and false positives). In this metric the three-class classifier outperformed the 5-class classifier with a precision of 87% and 78% respectively. This is largely due to the 4% improvement in classification of non-storm clouds by the three-class classifier. The improvement leads to less false positives than the five-class classifier even though the number of true positives actually decreased. The three-class classifier is better able to differentiate storm clouds and non-storm clouds as shown by the higher precision. Even though its accuracy on storm clouds is lower, for the purpose of SMICES the three-class classifier performs better on the tropical dataset.

The non-tropical dataset proved to be significantly more difficult to classify than the tropical dataset. Overall, the accuracy for the storm cloud classes (rainy anvil and convection core) decreased by about 2x. The non-storm class accuracy was relatively unchanged when compared to the tropical dataset for both classifiers. Overall, both the five-class and three-class classifier performed poorly, however, the three-class classifier identified roughly 5% of the convection

| | Clear, Thin Cirrus, and Cirrus | Rainy Anvil | Convection Core |
|---|---|---|---|
| Clear, Thin Cirrus, and Cirrus | 44,821<br>82.22% | 9,552<br>17.52% | 139<br>0.25% |
| Rainy Anvil | 3,199<br>16.87% | 15,418<br>81.31% | 344<br>1.81% |
| Convection Core | 173<br>22.09% | 408<br>52.11% | 202<br>25.80% |

**(a) 5 class labelled data**

| | Clear, Thin Cirrus, and Cirrus | Rainy Anvil | Convection Core |
|---|---|---|---|
| Clear, Thin Cirrus, and Cirrus | 45,560<br>83.08% | 9,124<br>16.64% | 152<br>0.28% |
| Rainy Anvil | 2,487<br>13.27% | 15,899<br>84.85% | 351<br>1.87% |
| Convection Core | 146<br>21.38% | 349<br>51.10% | 188<br>27.53% |

**(b) 3 class labelled data**

**Table 4 Confusion Matrix of the linear SVC classifier on the tropical dataset, weights balanced**

core clouds correctly compared to 12% in the five-class classifier, but did classify rainy anvil clouds roughly 5% better (Table 3). Both classifiers identified storm clouds at roughly the same accuracy of 42%. While this is not good, it is 2x more accurate than a random selection of pixels. The precision of the classifiers on identifying storm clouds also decreased with the three-class classifier performing better. The three-class and five-class classifiers generated a precision of 60% and 48% respectively.

Overall, training the RDF on five-class and three-class non-tropical data had different effects than the tropical dataset. While the three-class classifier outperformed in accuracy for non-storm, underperformed in accuracy for rainy anvil pixels, and tied in accuracy for convection core pixels in the tropical dataset, it underperformed the five-class classifier in accuracy of convection core pixels and rainy anvil pixels while outperforming in accuracy for non-storm pixels in the non-tropical dataset. Despite the changes in accuracy, the precision of the three-class classifier on storm clouds outperformed the five-class classifier in both datasets.

### B. Support Vector Machine (SVM)

The support vector machine used was the linear support vector classifier (linear SVC) from Scikit Learn. This classifier is suggested if the dataset contains greater than tens of thousands of data points by Scikit Learn. The linear SVC is effectively an support vector machine with a linear kernel.

Table 4 demonstrates the accuracy of the SVM classifiers with a linear kernel with the tropical dataset. The

| | Clear, Thin Cirrus, and Cirrus | Rainy Anvil | Convection Core |
|---|---|---|---|
| Clear, Thin Cirrus, and Cirrus | 3,464,133<br>89.15% | 307,787<br>7.92% | 113,775<br>2.93% |
| Rainy Anvil | 403,130<br>57.14% | 253,696<br>35.96% | 48,740<br>6.91% |
| Convection Core | 199,433<br>75.58% | 40,114<br>15.20% | 24,332<br>9.22% |

**(a) 5 class labelled data**

| | Clear, Thin Cirrus, and Cirrus | Rainy Anvil | Convection Core |
|---|---|---|---|
| Clear, Thin Cirrus, and Cirrus | 3,438,367<br>88.72% | 262,381<br>6.77% | 174,655<br>4.51% |
| Rainy Anvil | 354,782<br>61.06% | 162,209<br>27.92% | 64,088<br>11.03% |
| Convection Core | 273,907<br>68.71% | 57,672<br>14.47% | 67,079<br>16.83% |

**(b) 3 class labelled data**

**Table 5 Confusion Matrix of the linear SVC classifier on the non-tropical dataset, weights balanced**

three-class SVM outperformed its five-class counterpart in every class by 1-3%. The higher performing classifier performed well in the non-storm and rainy anvil class, returning 83% and 85% accuracy respectively, and poorly in the convection core class, returning 27% accuracy. When looking at the non-storm and storm accuracies, the three class SVM achieves a good distinction with an 83% accuracy in non-storm and an 84% accuracy in storms. The SVM classifiers performed similarly in precision as well with the three-class and five-class classifiers scoring 64% and 63% respectively.

The three-class classifier performed similarly to the five class random decision forest. The SVM classified the convection core class 4% higher and the rainy anvil class 3% higher than the five-class RDF. However, it also misclassified convection core clouds as non-storm clouds 9% more than the random decision forest. When comparing the non-storm and storm accuracies, the three-class SVM is 3% more accurate than the five-class random decision forest in storms, but 5% less accurate in non-storms. The RDF significantly outperforms the SVM classifier in precision with its highest precision of 87% compared to the best SVM precision of 64%. This is due to the SVM having a lower accuracy in non-storm clouds, which increases false positives. The RDF classifier outperformed the SVM on the tropical dataset due to a higher storm cloud precision.

The SVM classifier performed worse in the storm classes on the non-tropical dataset than it did on the tropical dataset. Both the five-class and three-class classifiers accurately identified 89% of the non-storm clouds, but their classification for the storm class classification were 38% and 36% respectively. While both classifiers achieved a similar

| | Clear, Thin Cirrus, and Cirrus | Rainy Anvil | Convection Core |
|---|---|---|---|
| Clear, Thin Cirrus, and Cirrus | 47,188<br>74.76% | 15,715<br>24.90% | 217<br>0.34% |
| Rainy Anvil | 736<br>7.45% | 8,926<br>90.33% | 220<br>2.23% |
| Convection Core | 269<br>21.45% | 737<br>58.77% | 248<br>19.78% |

**(a) 5 class labelled data**

| | Clear, Thin Cirrus, and Cirrus | Rainy Anvil | Convection Core |
|---|---|---|---|
| Clear, Thin Cirrus, and Cirrus | 35,553<br>86.66% | 5,381<br>13.12% | 91<br>0.22% |
| Rainy Anvil | 12,426<br>39.35% | 18,886<br>59.81% | 266<br>0.84% |
| Convection Core | 214<br>12.95% | 1108<br>67.03% | 331<br>20.02% |

**(b) 3 class labelled data**

**Table 6 Confusion Matrix of the Gaussian Naïve Bayes Classifier on the tropical dataset**

storm classification, the three class classifier performed better on the convection core class and worse on the rainy anvil class than the five-class classifier (Table 5. This pattern is opposite the relationship of the RDF classifers, where the three-class classifier was worse at convection core classification. The precision of both classifiers on storm clouds also decreased. The three-class and five-class SVM classifiers scored a precision of 47% and 45% respectively. Overall, due to the poorer precision and accuracy of storm clouds than the RDF, the SVM classifier did not perform as well.

### C. Gaussian Naïve Bayes

The Gaussian Naïve Bayes classifier used is from Scikit learn. The likelihood of each feature is assumed to be Gaussian.

Table 6 contains the results of the Gaussian Naïve Bayes classifier when trained on the tropical dataset. Both classifiers only reached an accuracy of 20% within the convection core class, however, the five class classifier misclassified more convection core clouds as non-storm than the three-class classifier. The performance on the rainy anvil clouds is very different, with the five-class classifier correctly identifying 90% of the clouds compared to 60% with the 3-class classifier. The non-storm class was better classified by the three-cloud classifier. Analyzing the storm vs. non-storm accuracies show that the five-class classifier is very accurate at identifying storm clouds, identifying 91% of the storms accurately. However, this comes at the cost of accuracy in the non-storm class where the classifier only reached an accuracy of 75%. This is significantly below the three-class GNB and other classifiers which were able to reach around

| | Clear, Thin Cirrus, and Cirrus | Rainy Anvil | Convection Core |
|---|---|---|---|
| Clear, Thin Cirrus, and Cirrus | 3,393,184<br>92.03% | 187,456<br>5.08% | 106,523<br>2.89% |
| Rainy Anvil | 599,488<br>56.15% | 394,857<br>36.98% | 73,349<br>6.87% |
| Convection Core | 74,021<br>73.81% | 19,290<br>19.24% | 6,972<br>6.95% |

**(a) 5 class labelled data**

| | Clear, Thin Cirrus, and Cirrus | Rainy Anvil | Convection Core |
|---|---|---|---|
| Clear, Thin Cirrus, and Cirrus | 3,269,539<br>92.15% | 172,290<br>4.86% | 106,192<br>2.99% |
| Rainy Anvil | 745,390<br>60.34% | 413,333<br>33.46% | 76,572<br>6.20% |
| Convection Core | 51,743<br>72.04% | 16,012<br>22.29% | 4,069<br>5.67% |

**(b) 3 class labelled data**

**Table 7 Confusion Matrix of the Gaussian Naïve Bayes Classifier on the non-tropical dataset**

90%. There is a large disparity in the precision of storm clouds between the two classifiers. The three-class classifier achieved a 79% precision compared to 39% with the five-class classifier. Overall both classifiers have problems. The three-class classifier has a low storm cloud accuracy. The five-class classifier has a high accuracy of storm clouds, but also has a very low precision.

Table 7 shows the performance of the GNB classifiers over the non-tropical dataset. Both classifiers performed similarly in every category, with the 5-class classifier performing slightly better in the convection core and rainy anvil classes. The accuracies in the convection core and rainy anvil classes were lower than the accuracies achieved by the SVM and RDF classifiers. When analyzing the classifiers over the two class problem of non-storm and storm, the five-class classifier performed slightly better than the three-class classifier in the storm class. Precision in storms decreased with the three-class classifier while increasing for the five-class classifier. This resulted in similar precision scores of 65% and 63% for the three-class and five-class classifiers respectively. When comparing the two-class accuracies of the five-class GNB to the best performing non-tropical classifier, the three-class RDF, it performed identically. Both classifiers achieved around 92% accuracy in the non-storm class and 42% accuracy in the storm class. The GNB classifier outperformed the RDF in precision however, with a best score of 65% compared to 60%.

| | Clear, Thin Cirrus, and Cirrus | Rainy Anvil | Convection Core |
|---|---|---|---|
| Clear, Thin Cirrus, and Cirrus | 46,202<br>95.87% | 1,889<br>3.92% | 102<br>0.21% |
| Rainy Anvil | 13,044<br>51.40% | 12,043<br>47.46% | 290<br>1.14% |
| Convection Core | 148<br>21.57% | 389<br>56.71% | 149<br>21.72% |

**(a) 5 class labelled data**

| | Clear, Thin Cirrus, and Cirrus | Rainy Anvil | Convection Core |
|---|---|---|---|
| Clear, Thin Cirrus, and Cirrus | 45,752<br>94.93% | 2,377<br>4.93% | 64<br>0.13% |
| Rainy Anvil | 11,700<br>46.11% | 13,505<br>53.22% | 170<br>0.67% |
| Convection Core | 118<br>17.15% | 436<br>63.37% | 134<br>19.48% |

**(b) 3 class labelled data**

**Table 8 Confusion Matrix of the ANN on the tropical dataset**

### D. Neural Networks

In addition to standard machine learning classifiers, we explored the use of simple neural networks using Keras [27] and TensorFlow [28].

#### *1. Feed Forward Artificial Neural Network (ANN)*

We pass each 8-band pixel into a feed forward ANN with 2 hidden layers, 32 nodes in each hidden layer, a dropout rate of 0.1, and a softmax activation at the final layer. We used ADAM [29] as our optimizer. Model performance was not affected by small changes in these parameters. The model output is a vector with probability of each cloud-type class, and we take the Argmax to get the predicted class.

The ANN performance over the tropical dataset was relatively poor compared to the past classifiers. While both the five-class and three-class classifier achieved around 95% accuracy in the non-storm class, they struggled to identify both the convection core and rainy anvil clouds (Table 8). Even in the two class problem neither classifier was able to identify storm clouds with 60% accuracy. The extremely high accuracy in the non-storm class did result in a very high precision in the storm class with the scores of 85% and 87% in the three-class and five-class classifiers respectively

The ANN was not effective on the non-tropical dataset, and labelled all pixels as the non-storm class during training and validation. This is probably due to the larger skew of the non-tropical dataset towards the non-storm class.

| | Clear, Thin Cirrus, and Cirrus | Rainy Anvil | Convection Core |
|---|---|---|---|
| Clear, Thin Cirrus, and Cirrus | 43,893<br>91.08% | 4190<br>8.69% | 110<br>0.23% |
| Rainy Anvil | 5,926<br>23.35% | 19,000<br>74.88% | 449<br>1.77% |
| Convection Core | 83<br>12.06% | 405<br>58.87% | 200<br>29.07% |

**(a) 5 class labelled data**

| | Clear, Thin Cirrus, and Cirrus | Rainy Anvil | Convection Core |
|---|---|---|---|
| Clear, Thin Cirrus, and Cirrus | 43,871<br>91.03% | 4,221<br>8.76% | 101<br>0.21% |
| Rainy Anvil | 5,650<br>22.26% | 19,243<br>75.83% | 485<br>1.91% |
| Convection Core | 46<br>6.72% | 434<br>63.36% | 205<br>29.93% |

**(b) 3 class labelled data**

**Table 9 Confusion Matrix of the CNN on the tropical dataset**

*2. Convolutional Neural Network (CNN)*

We also test a single-pixel CNN model. Our architecture uses 3 stacked 1-d convolutional layers, where the convolutions are applied over the bands. Due to the small input size, we did not apply pooling. 6, 12, and 24 filters were used for each convolutional layer respectively, and a rectified linear unit (relu) activation was used. We found a convolutional filter shape of 1x3 to be optimal. The last layers of our net included two fully connected layers with 32 nodes, a dropout of 0.1, and a final sofmax activation. ADAM optimizer was used to train the network.

Table 9 show that the three-class and five-class classifiers perform almost identically on the tropical dataset. Throughout the results the accuracy only changes by at most 1%. The CNN classifiers were able to achieve the highest accuracy in the convection core class, reaching 30%. They also performed well in the rainy anvil class with an accuracy of 75% and achieving 91% in the non-storm cloud class. These results also carry to the two class problem where they found storm clouds with 77% accuracy. Both classifiers have the same precision score for storm clouds of 82%. Overall, the 5-class RDF classifier was able to identify storm clouds 7% better due to a better classification of the rainy anvil clouds. The CNN does classify non-storm clouds slightly better than the five-class RDF, but underperforms when compared to the three-class RDF.

The CNN experiences a similar outcome to the ANN when trained on the non-tropical dataset. Table 10 shows that the heavier skew towards non-storm clouds caused the classifier to classify the majority of the test dataset as non-storm. For future work, we plan to upsample the minority classes when training our neural networks.

| | Clear, Thin Cirrus, and Cirrus | Rainy Anvil | Convection Core |
|---|---|---|---|
| Clear, Thin Cirrus, and Cirrus | 3,348,548<br>90.82% | 307,505<br>8.34% | 30,902<br>0.84% |
| Rainy Anvil | 539,227<br>59.88% | 322,980<br>35.86% | 38,341<br>4.26% |
| Convection Core | 228,529<br>85.39% | 30,029<br>11.22% | 9,079<br>3.39% |

**(a) 5 class labelled data**

| | Clear, Thin Cirrus, and Cirrus | Rainy Anvil | Convection Core |
|---|---|---|---|
| Clear, Thin Cirrus, and Cirrus | 3,349,638<br>90.85% | 306,147<br>8.30% | 31,156<br>0.85% |
| Rainy Anvil | 559,743<br>62.15% | 309,828<br>34.40% | 30,990<br>3.44% |
| Convection Core | 228,680<br>85.44% | 32,838<br>12.27% | 6120<br>2.29% |

**(b) 3 class labelled data**

**Table 10 Confusion Matrix of the CNN on the non-tropical dataset**

## VIII. Robustness of Learned Classifiers to Input Noise

The analysis to this point has been using clean radiance values of the simulated radiometer. In flight, however, we cannot expect perfect data. Instead some noise is expected. In flight we expect to see Gaussian noise that is independent along each radiance band. Along the bands of Tb380 we expect to find around 5 kelvin of noise and 1 kelvin of noise along the remaining bands. We do not expect noise to change based on the viewing angle of the radar, however the transmitted power will be less due to the longer travel of the signal. We generate random values from a Gaussian distribution with a mean of 0 and a standard deviation of the expected noise for each band, and add this independently to each band to create the noisy dataset. The impact of noise was tested using the three-class RDF classifier due to its good performance across both datasets. The impact of noise will be determined by the total decrease in accuracy between the classifier tested on the clean held-out test set, and on the same testset, but with noise applied.

1) Tropical dataset: RDF classifier with a max depth of 14, 32 trees, and weights equalized was trained on the first eight images of the tropical dataset. It was then tested on the remaining five images after expected noise was applied to those data values.
2) Non-tropical dataset: RDF classifier with a max depth of 14, 32 trees, and weights equalized was trained on the first image of the non-tropical dataset (10 image cutouts). It was then tested on the second image (10 image cutouts) after expected noise was applied noise was applied to those data values.

Over the tropical dataset the accuracy of the three-class RDF decreased by 4% due to the expected noise in flight.

This proves that the RDF classifier tested is relatively robust to noise in the tropical dataset. When applied to the non-tropical dataset the accuracy of the RDF decreased by 2% the accuracy due to expected noise in flights.

## IX. Smart Targeting Background

The SMICES problem is a continuous online planning problem implemented as an orbiting satellite where there is no set end to the imaging of the clouds. The parameters include an orbiting altitude of 400km and the setup is designed to allow its radar to slew 15° from nadir in all directions. The slewing is assumed to have instantaneous electronic movement. The radar is capable of targeting an area of roughly 4x4km whenever it is turned on. The power constraints of the vehicle mean that the targeting should reach a 20% duty cycle at any given time over the course of the flight. If these constraints were applied to current satellite targeting techniques, the result would be randomly sampling 20% of the clouds at nadir under the satellite. Identification of the clouds occurs within the radiometer's range, which sweeps at 45° ahead of nadir. This sweep covers 60° around the satellite and therefore covers more ground than the radar is capable of viewing.

When modelling the radar's viewpoint, shown in Figure 4, we focus on the knowledge window, which is defined as the area from the back of the radar's reachability to the front of the radiometer sweep. The window is only as wide as the radar's movement since we are unable to target any clouds outside of this range. We investigate smart targeting in the context of this model.

Within our knowledge window it is necessary to define the priority of the different clouds that will be viewed. The classifier onboard is capable of distinguishing between five different cloud types, clear sky, thin cirrus, cirrus, rainy anvil (RA), and convection core (CC), in order of increasing interest. In particular, the rainy anvil and convection core clouds are the most important since they make up storms. The scientists also want to make sure that the radar is collecting data from all of the cloud types it passes over since there can be useful information in the non-storm clouds as well. To accomplish this, whenever an algorithm is not targeting rainy anvil and convection core clouds, the radar will be taking a random sampling of the clouds under nadir. To simplify, the algorithms will be created to prioritize convection core clouds, followed by rainy anvil clouds, and then a random sample of the clouds under nadir.

Multiple algorithms were created with an increasing field of view to demonstrate the improvements gained by utilizing more of the information available to the instrument. The random algorithm serves as the baseline comparison as it is representative of the results we would receive without targeting any clouds. The only algorithms that utilize the entirety of the knowledge window are the windowed smart and greedy path, which plan out the projected usage of power beyond the immediately reachable pixels.

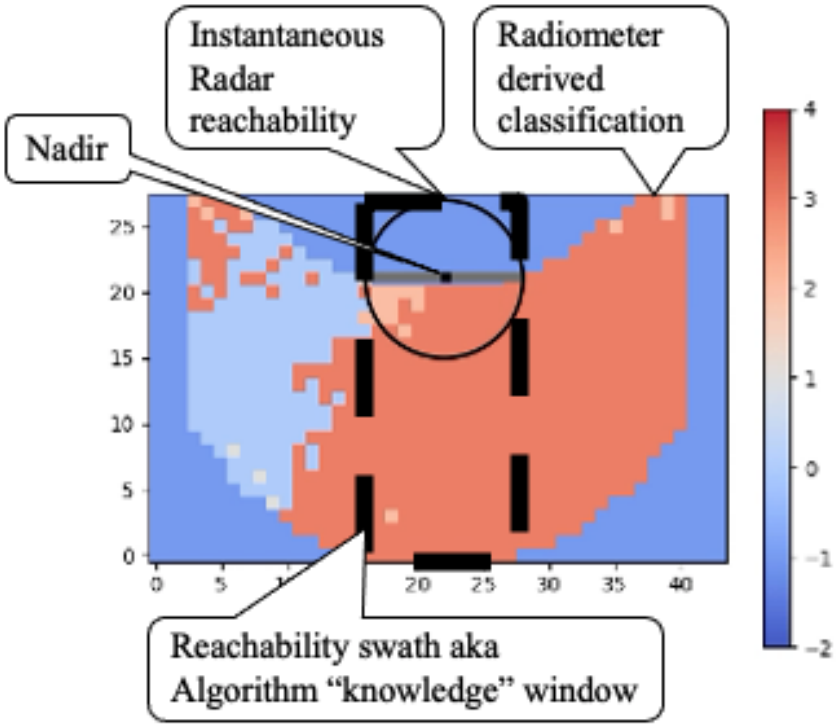


**Fig. 4 Example of the field of view of the satellite. Identified storms are represented by each color: 4.0 = convection core, 3.0 = rainy anvil, 2.0 = cirrus, 1.0 = thin cirrus, 0.0 = clear, -2 = not analyzed. Radar's View: black circle, Nadir: black square, Knowledge Window: dotted rectangle**

## X. Smart Targeting Algorithms

The duty cycle of the algorithms is dictated by the state of charge (SOC) of its system. In order to ensure that the SMICES duty cycle of 20% is maintained the SOC is decreased by 4% when the algorithms analyze a pixel, and the SOC is increased by 1% when the radar is left off. The simulation begins with 0% power.

Throughout this Section there are figures displaying how each algorithm would perform in a given knowledge window. In each figure nadir is represented by the black square and the radar's view is displayed by the circle. The different cloud types are represented on the color bar such that 4.0 = convection core, 3.0 = rainy anvil, 2.0 = cirrus, 1.0 = thin cirrus, 0.0 = clear. These values also represent the reward given for analyzing each cloud type. The state of charge for every figure is assumed to be 50%.

### A. Random Targeting Algorithm

The random algorithm (Figure 5 and Algorithm 1) targets the pixel under nadir 20% of the time to ensure that it meets the energy requirements for SMICES. It is representative of most targeting methods on current Earth Science satellites today. Its random nature means that it is indifferent to the clouds it is flying over and will miss some important clouds.

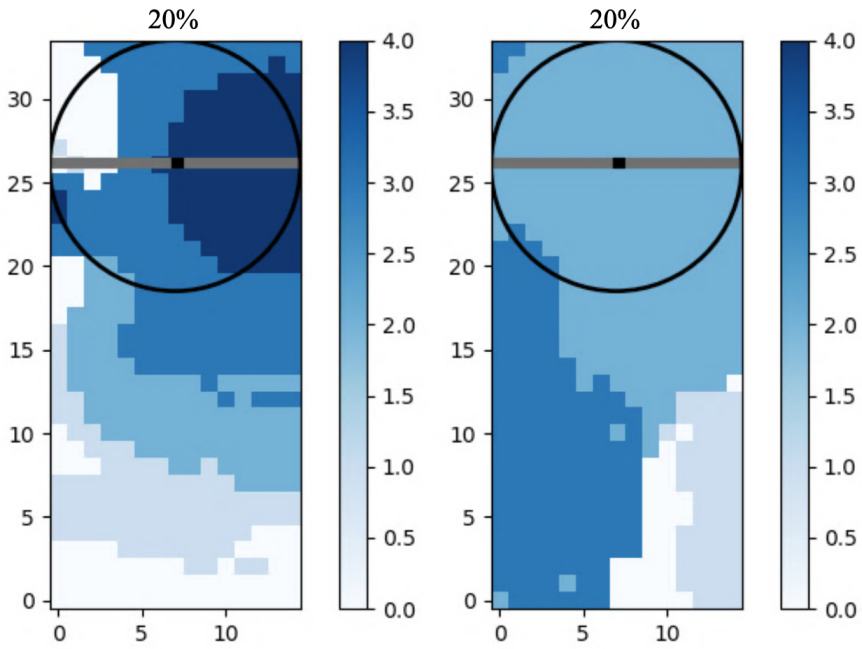


**Fig. 5 Chance that the radar analyzes the pixel under the random algorithm**

**Algorithm 1:** Random Algorithm

**output** Results: array of analyzed pixel values

⋮

**input** : Results: array of analyzed pixel values, Picture: knowledge window of the simulation

$i \leftarrow randomvalue(0 < i < 1)$

**if** $i \leq .2$ **then**

  $results \leftarrow$ value of pixel at nadir

  `// add the value of pixel under nadir to results`

**end**

**return** *Results*

### B. On/Off Targeting Algorithm

The on/off algorithm (Figure 6 and Algorithm 2) improves the random algorithm by controlling when the radar is turned on. It utilizes the system's current energy state and the cloud type under nadir to determine when the radar is turned on instead of using a random generator. This allows the system to save energy when there are no interesting clouds, and use the stored energy when there are. It also mimics random by taking the value under nadir when the SOC is high.

### C. Lateral Targeting Algorithm

The lateral algorithm (Figure 7 and Algorithm 3) improves on the on/off algorithm by allowing the radar to analyze pixels along the cross-path direction. This is symbolized in the knowledge window graphic by the band that crosses nadir. The two important factors in determining when the radar is turned on is the state of charge of the vehicle and the

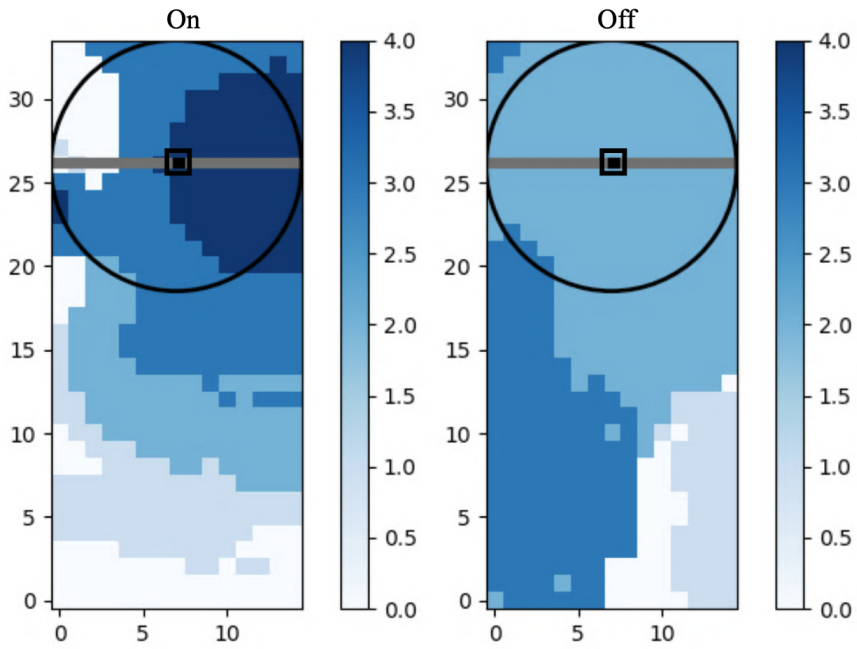


**Fig. 6 Left: radar is on because a convection core is under nadir. Right: radar is off because a cirrus cloud is under nadir. The radar turns on when it sees good pixels under nadir in the on/off algorithm**

best pixel along the lateral band. This is resolved by searching for the highest valued cloud with a tiebreaker going to the pixel that is closest to nadir. If two pixels are equidistant from nadir, the pixel to the left is chosen.

**Algorithm 2:** On/off Algorithm

**output** Results: array of analyzed pixel values, power: power state of the system, sample: boolean to sample if there are no high priority targets

**input** : Results: array of analyzed pixel values, Picture: knowledge window of the simulation, Power: power state {0-100}, Sample

```
if power > 60 then
    results ← value of pixel under nadir
    power ← power − 4
    sample ← True
else if power > 40 and sample then
    results ← value of pixel under nadir
    power ← power − 4
    sample ← True
else if power > 4 then
    if Pixel under nadir == (CC or RA) then
        results ← value of pixel under nadir
        power ← power − 4
    else
        results ← value of radar turned off
        power ← power + 1
    end
    sample ← False
else
    results ← value of radar turned off
    power ← power + 1
    sample ← False
return Results
```

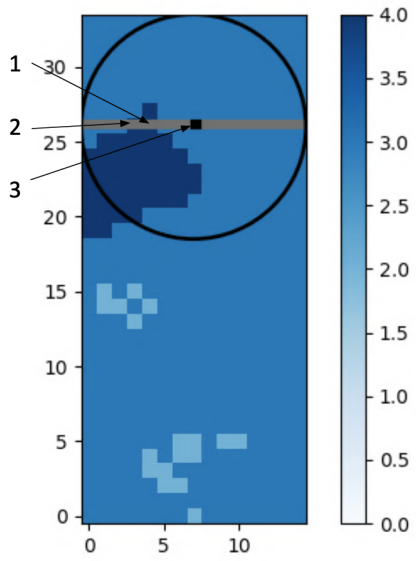


**Fig. 7 Top three prioritized pixels based on the lateral algorithm**

### D. Smart Targeting Algorithm

The smart algorithm (Figure 8 and Algorithm 4) expands its view along the path of the satellite to include the entirety of the radar's reachability. This area is signified by the black circle in the graphic. When deciding which pixel to analyze for a given time step, the smart algorithm follows steps similar to the lateral algorithm. The state of charge determines which cloud types are able to be analyzed, and a search inside of the radar's reachability finds the highest valued cloud with a tiebreaker going to the pixel that is closest to nadir. In the case of two equidistant pixels, the pixel that is laterally closer to nadir is chosen, and then if the tie remains the leftmost pixel is chosen.

### E. Windowed Smart Targeting Algorithm

The windowed smart algorithm (Figure 9 and Algorithm 5) expands its view to include the entire knowledge window of the simulation. This increased view now exceeds the radar's reachability, meaning that the algorithm is able to account for future clouds along the radar's path. The algorithm first calculates how many clouds can be analyzed based on the current state of charge. It then counts the number of convection core and rainy anvil clouds present within the knowledge window. The power is then allocated for all of the convection core pixels, followed by the rainy anvil pixels, and then any leftover power is reserved as free. The highest valued pixel within the radar's view that has allocated power is imaged. The tiebreaker for priority follows the same logic as the smart algorithm. The pixel under nadir is imaged if neither a convection core or rainy anvil pixel are within the radar's view, there is free power, and there is a sufficient SOC.

**Algorithm 3:** Lateral Algorithm

**output** Results: array of analyzed pixel values, power: power state of the system {0-100}, sample: boolean to :
sample if there are no high priority targets
**input** :Results, Picture: knowledge window of the simulation, Power, Sample

```
clouds ← pixels that make up lateral band across nadir within radar's view
best ← lat_search(clouds) // Returns the best pixel in the lateral field of view that is closest to nadir
if power > 60 then
    if best == RA or CC then
        results ← value of best
    else
        results ← value of pixel under nadir
    end
    power ← power − 4
    sample ← True
else if power > 40 and sample then
    if best == RA or CC then
        results ← value of best
    else
        results ← value of pixel under nadir
    end
    power ← power − 4
    sample ← True
else if power > 4 then
    if best == (CC or RA) then
        results ← value of best
        power ← power − 4
    else
        results ← value of radar turned off
        power ← power + 1
    end
    sample ← False
else
    results ← value of radar turned off
    power ← power + 1
    sample ← False
return Results, Power, Sample
```

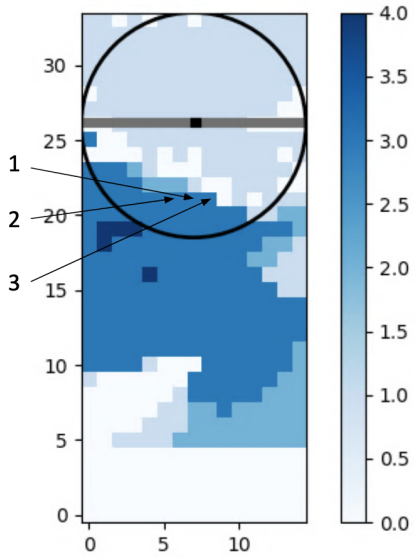


**Fig. 8 Top three prioritized pixels based on the smart algorithm**

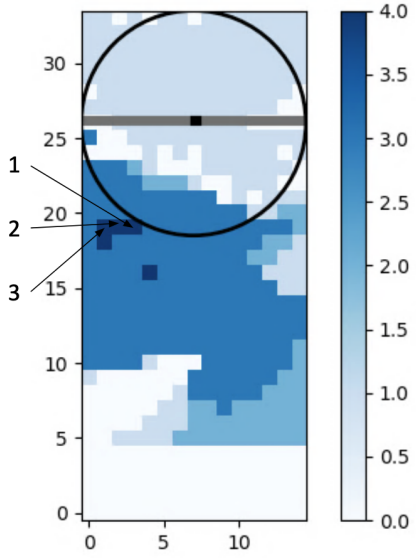


**Fig. 9 Top three prioritized pixels based on the windowed smart algorithm**

**Algorithm 4:** Smart Algorithm

**output** Results: array of analyzed pixel values, power: power state of the system {0-100}, sample: boolean to :
sample if there are no high priority targets
**input** :Results, Picture: knowledge window of the simulation, Power, Sample

```
radar_view ← pixels that make up radar's range of possible targets
best ← smart_search(radar_view, view_radius) // Returns the best pixel in the radar's field of view that is closest to nadir
if power > 60 then
    if best == RA or CC then
        results ← value of best
    else
        results ← value of pixel under nadir
    end
    power ← power − 4
    sample ← True
else if power > 40 and sample then
    if best == RA or CC then
        results ← value of best
    else
        results ← value of pixel under nadir
    end
    power ← power − 4
    sample ← True
else if power > 4 then
    if best == (CC or RA) then
        results ← value of best
        power ← power − 4
    else
        results ← value of radar turned off
        power ← power + 1
    end
    sample ← False
else
    results ← value of radar turned off
    power ← power + 1
    sample ← False
return Results, Power, Sample
```

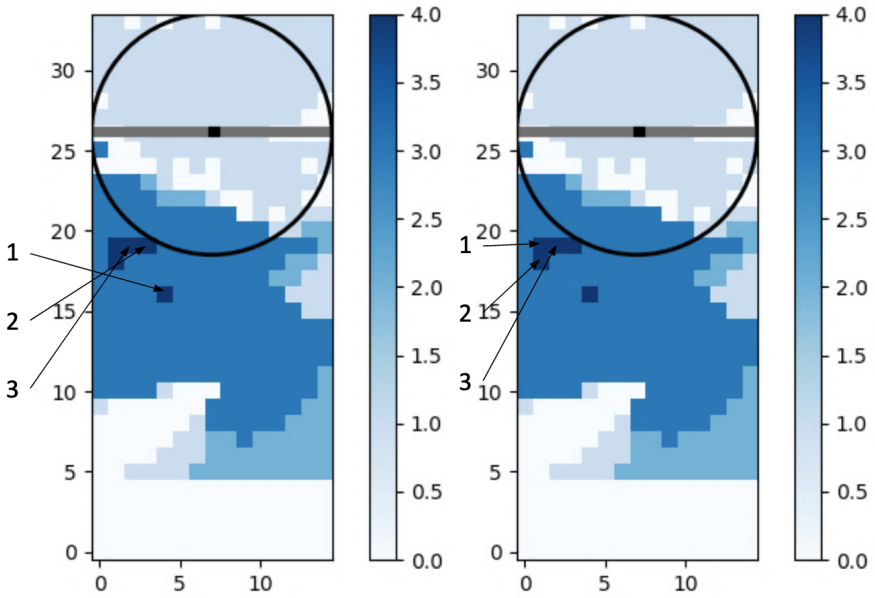


**Fig. 10 Left: Top three prioritized pixels based on the greedy path algorithm. Right: Top three prioritized pixels based on the greedy path wide algorithm**

### F. Greedy Path Targeting Algorithm

Greedy path (Figure 10 and Algorithm 6) improves upon the windowed smart algorithm by ranking the priority of each convection core and rainy anvil pixel in the knowledge window. The algorithm begins by collecting the locations of all these pixels and calculating the available radar cycles based on the state of charge. Once collected, the two pixel types are sorted independently by their lateral distance to nadir. This means that a newly scanned convection core pixel that will eventually cross nadir will have a higher priority than an off-nadir convection core pixel within the radar's view. The sorted list of rainy anvil pixels is then concatenated to the end of the sorted list of convection core pixels to create a priority queue. Greedy path then assigns one radar cycle to the highest priority pixel and checks if it is within the radar's view. If it is viewable, the pixel is analyzed. Otherwise, it continues until the free cycles run out or the priority queue ends. If free cycles are left over after the end of the priority queue and the SOC is sufficient, the algorithm will analyze nadir.

Greedy path has two variations, greedy path and greedy path wide. The difference is in how the pixels are prioritized. Clouds that run directly under nadir are within the radar's field of view for significantly longer than the clouds that run just under the edge. Greedy path wide takes this into account and always chooses the cloud that is laterally farther away from nadir if there is a tie in priority. The logic is that it will be able to analyze more high priority clouds by choosing the ones that are within its field of view for the shortest time. In contrast, the normal greedy path targets the cloud that is laterally closer to nadir in the case of a tie. In the case that two pixels are laterally equidistant, the algorithms select the leftmost pixel.

**Algorithm 5:** Windowed Smart Algorithm

**output** Results: array of analyzed pixel values, power: power state of the system {0-100}, sample: boolean to :
sample if there are no high priority targets
**input** :Results, Picture: knowledge window of the simulation, Power, Sample

$free\_cycles \leftarrow$ power / 4
// free_cycles is set to the total number of times the radar can be turned on at the current power state
$storms \leftarrow$ dictionary mapping the storm types to their occurrence in the knowledge window
$best\_cc, best\_ra \leftarrow$ smart_search(radar_view, view_radius) // Returns the best CC pixel and the best RA pixel in the radar's field of view that are closest to nadir
$radar\_view \leftarrow$ pixels that make up radar's range of possible targets
**if** $power > 60$ **then**
    sample = True
**else if** $power < 40$ **then**
    sample = False
// set the sampling variable
**if** *CC in storms* **then**
    $cc \leftarrow$ number of CC pixels in Picture
**if** *RA in storms* **then**
    $ra \leftarrow$ number of RA pixels in Picture
**if** $free\_cycles \leq cc$ **then**
    $cc \leftarrow$ free_cycles
    $free\_cycles \leftarrow 0$
**else**
    $free\_cycles \leftarrow$ free_cycles - cc
**end**
// set cc to the total number of CC pixels that can be pictured
**if** $free\_cycles \leq ra$ **then**
    $ra \leftarrow$ free_cycles
    $free\_cycles \leftarrow 0$
**else**
    $free\_cycles \leftarrow$ free_cycles - ra
**end**
// set ra to the total number of RA pixels that can be pictured
**if** $cc > 0$ *and best_cc exists* **then**
    **if** $cc > 0$ **then**
        $results \leftarrow$ value of best_cc exists
        $power \leftarrow power - 4$
    **else if** $ra > 0$ *and best_ra exists* **then**
        $results \leftarrow$ value of best_ra
        $power \leftarrow power - 4$
    **else if** $free\_cycles > 0$ *and sample* **then**
        $results \leftarrow$ value of pixel under nadir
        $power \leftarrow power - 4$
    **else**
        $results \leftarrow$ value of radar turned off
        $power \leftarrow power + 1$
**return** *Results, Power, Sample*

**Algorithm 6:** Greedy Path Algorithm

```
output Results: array of analyzed pixel values, power: power state of the system {0-100}, sample: boolean to
:
        sample if there are no high priority targets
input :Results, Picture: knowledge window of the simulation, Power:, Sample
free_cycles ← power / 4
    // free_cycles is set to the total number of times the radar can be turned on at
 the current power state
if power > 60 then
    sample = True
else if power < 40 then
    sample = False
                                                                    // set the sampling variable
cc_pixels ← empty array
ra_pixels ← empty array
for Each pixel in Picture do
    if pixel is CC then
        cc_pixels ← location of the pixel
    if pixel is RA then
        ra_pixels ← location of the pixel
end
cc_pixels ← sort(cc_pixels)
ra_pixels ← sort(ra_pixels)      // sort cc_pixels and ra_pixels in order of how laterally
 close they are to nadir
     // for greedy path wide the sorting would be on how laterally far they are from
 nadir
interesting_pixels ← cc_pixels + ra_pixels
checks ← 0
undecided ← True
while checks < free_cycles and undecided do
    if checks <length of interesting_pixels then
        point ← interesting_pixels[checks] if point is within the radar's view then
            undecided ← False
            results ← value of pixel at point
            power ← power − 4
    else
        undecided ← False
        if sample then
            results ← value of pixel under nadir
            power ← power − 4
        else
            results ← value of radar turned off
            power ← power + 1
        end
    end
    checks ← checks + 1
end
if undecided then
    results ← value of radar turned off
    power ← power + 1
return Results, Power, Sample
```

| Battery State of Charge (SOC) (0-100%) | Decision Process |
|---|---|
| SOC > 60% | nadir_sampling := ON;<br>If convection core reachable then observe convection core;<br>Else if rainy anvil reachable then observe rainy anvil;<br>Else observe nadir; |
| 60% > SOC > 40% | If convection core reachable then observe convection core;<br>Else if rainy anvil reachable then observe rainy anvil;<br>Else if nadir_sampling = ON then observe nadir; |
| 0% > SOC > 40% | If convection core reachable then observe convection core;<br>Else if rainy anvil reachable then observe rainy anvil;<br>nadir_sampling := OFF; |
| 0% = SOC | Do not sample |

**Table 11 Energy level thresholds for viewing pixels**

### G. Energy Level Thresholds

After the random algorithm the pixel selection is dictated by the energy level instead of a random generator. At full power the system should always be analyzing a cloud, but as the power drains the list of valid clouds to analyze becomes more constrained.

Table 11 highlights the thresholds used in the previously stated algorithms. When the state of charge is greater than 60%, the algorithms will either be view a convection core or rainy anvil pixel, or sample nadir. The sampling variable is also set to "on" in this threshold. If the power is between 40% and 60% and the sampling variable is on, then the algorithms perform the same as when the power is above 60%. If the variable is off, then the algorithms will only view convection core and rainy anvil pixels. Once the power drops below 40% the algorithms also only view rainy anvil and convection core pixels. The sampling variable is set to "off" at this threshold. The purpose of the sampling variable is to randomly sample nadir between the 60% and 40% SOC until high priority clouds come into view.

One important feature basing cloud selection on the available energy is its flexibility. The 60% and 40% decision boundaries are not set in stone, and can be changed to better fit a scientist's interest. If the boundaries were to be shifted down, then the algorithm would start randomly sampling more, and the results would be a more even distribution across different cloud types. Inversely, if the boundaries were each shifted up, then less random sampling would occur. This would cause the results to skew heavier to the convection core and rainy anvil clouds. Additionally, this algorithm can be adapted to focus on any types of identified classes such as cirrus and thin cirrus.

## XI. Smart Targeting Experimental Design

The tropical dataset described in Section 4 was used to evaluate the targeting algorithms. During the simulation the along track length is 1,785km and the across track length is 3,120km.

To prepare the images for the algorithms, each pixel is passed through a trained random decision forest that identifies

| | Off | Clear | Thin Cirrus | Cirrus | Rainy Anvil | Convection Core |
|---|---|---|---|---|---|---|
| Random (20% on) | 16,079 | 1,084 | 439 | 1,099 | 1,367 | 43 |
| | 79.95% | 26.88% | 10.89% | 27.26% | 33.90% | 1.07% |
| On/Off | 16,097 | 720 | 142 | 405 | 2,694 | 53 |
| | 80.04% | 17.94% | 3.54% | 10.09% | 67.12% | 1.32% |
| Lateral (1 DoF + on/off) | 16,094 | 482 | 111 | 153 | 3,088 | 183 |
| | 80.03% | 12.00% | 2.76% | 3.81% | 76.87% | 4.56% |
| Smart (1 DoF + on/off) | 16,092 | 352 | 95 | 86 | 3,107 | 379 |
| | 80.02% | 8.76% | 2.36% | 2.14% | 77.31% | 9.43% |
| Windowed Smart | 16,093 | 313 | 64 | 43 | 2,850 | 748 |
| | 80.02% | 7.79% | 1.59% | 1.07% | 70.93% | 18.62% |
| Greedy Path | 16,093 | 313 | 64 | 43 | 2,579 | 1,019 |
| | 80.02% | 7.79% | 1.59% | 1.07% | 64.19% | 25.36% |
| Greedy Path Wide | 16,094 | 411 | 53 | 80 | 2,415 | 1058 |
| | 80.03% | 10.23% | 1.32% | 1.99% | 60.12% | 26.34% |

**Table 12 The Table contains the results of running the algorithms over all 13 images of the dataset. The runs were organized by shifting nadir to the East by the diameter of the radar view after every run. Each image contains 13 runs. 20,111 timesteps were taken over the entire dataset. During each time step an algorithm is capable of analyzing one pixel or turning off the radar. Percentages inside shaded cells are normalized over the time the radar was on during the runs. Greedy path found 23.7x more convection core than random, 1.9x more rainy anvil than random.**

the type of cloud represented by the pixel. It labels CC pixels with 89% accuracy, RA pixels with 75% accuracy, and clear, thin cirrus, and cirrus pixels with 92% accuracy when the expected radiometer noise is incorporated. Clear, thin cirrus, and cirrus is considered once class because the algorithms never actively target any of those cloud types.

The targeting algorithms are then run over the classified image through non overlapping vertical paths from the top to the bottom of the image. This pathing is utilized because we do not have the data to use realistic flight paths. The first path is at the far left of the image so that the left side of the radar's view matches with the left hand border of the scene. The satellite is assumed to be centered over the middle pixel of the path. Once that run is complete the second run is shifted right by the radius of the radar's view so that no pixels overlap in between the runs. Each run progressively shifts right until it is impossible to fit another run without going over the boundary of the image or repeating pixels. The runs begin with nadir at the top of the image and end when nadir reaches the bottom.

We assume that the images are snapshots and do not change relative to the overflight time. Given the flight speed of 7.5km/s each run down the image only takes roughly four minutes. We are able to fit 13 runs per image, bringing the traversal time for an entire snapshot to roughly 52 minutes. The algorithms are able to analyze one pixel or turn off the radar during each time step of the simulation. In practice the "off" state is a stand-by mode where the most power consuming subsystems are put on hold, but not fully off. This means that we assume that the radar can be powered up and down instantaneously with no impact on data quality. Each timestep is the traversal time of one pixel, and given the

pixel size of 15km each timestep in the simulation represents roughly 2 seconds. It is important to note that this dataset contains a significantly higher number of storms than we would predict to see in a real flight. This is expected to skew the results of the algorithm towards greater performance since there are more storms to target.

For this simulation we are assuming that the radar is able to aim at the middle of every pixel within its field of view and count the value of every targeted pixel equally regardless of location. We also assume that the radiometer is collecting data at the 15km x 15km pixel size. The actual radiometer has a pixel size of 6km x 10km at the highest frequency and a lower resolution at lower frequencies. This study does not factor in the fact that the size of the radar's footprint changes as the radar targets off nadir.

## XII. Smart Targeting Result

Table 12 shows that over the course of the runs it is clear that the dynamic targeting delivers a significant increase in performance. In the on/off scenario there was a slight increase in the number of convection core pixels analyzed, and the number of rainy anvil pixels evaluated effectively doubled. The number of these pixels chosen continues to rise as the view of the algorithm increases to the size of the radar's view. These increases can be explained by the improved possibility of seeing high priority pixels as well as a longer amount of time to analyze the pixels as the field of view expands along the radar's path.

A more interesting change is found when the algorithm's field of view expands from just the radar's view in the smart algorithm to the entire knowledge window in the windowed smart algorithm. When the two algorithms are compared the windowed smart algorithm views less rainy anvil pixels and more convection core pixels than the smart algorithm.

This is largely attributed to the distribution of clouds across the sky. Clouds are not randomly distributed. Instead, they cluster around storms. In particular, the convection core clouds serve as the center of most storms with rainy anvil clouds surrounding the center. The clustering means that the most important clouds for the algorithms are always found together. This problem is then exacerbated by the fact that the most scientifically interesting clouds, convection cores, are inside of large clusters of also highly prioritized rainy anvil clouds.

The distribution is a problem due to the energy constraints of the SMICES radar. Because both cloud types are scientifically significant, the targeting algorithms use any remaining power to analyze either of these clouds when they fall within the radar's field of view. This can lead to all of the available power being used to target the surrounding rainy anvil clouds and leave no power once the convection core clouds come into view. As the knowledge window increases, the algorithms are able to budget their remaining power by allocating resources to the future convection core clouds that will eventually enter the radar's view. The windowed smart algorithm is able to budget its power to save for future convection core pixels while the smart algorithm is not. This budgeting of power explains the increase in convection core performance at the expense of rainy anvil.

This feature of budgeting power is further developed in the greedy path algorithms, which selectively choose only

the best rainy anvil and convection core pixels out of its knowledge window. The development is reflected in the results which show that the greedy path algorithm analyzes a decreased number of rainy anvil pixels and an increased number of convection core pixels when compared to the windowed smart algorithm.

These results also demonstrate that the wide approach continues this trend and analyzes slightly more convection core pixels at the cost of some rainy anvil pixels. Unfortunately, looking at clouds that are farther off of nadir returns worse data due to the increased distance to the imaging target. More analysis on how off nadir the analyzed clouds are and how severely the image quality degrades per angle off nadir is necessary to deduce whether this algorithm is actually superior. For this paper comparisons will be made to the greedy path algorithm since it prioritizes the clouds closer to nadir.

## XIII. Discussion

Multiple classifiers were able to accurately distinguish between the two-class problem in the tropical dataset. It is important to understand that even less accurate classifiers can be used to significantly increase yield of the SMICES mission concept. In Table 13a the first row shows the distribution of pixels acquired if sampling pixels at random (e.g. not using any classification). Row two shows the distribution if we sample pixels the 5-class RDF classifies as Rainy Anvil. Note that in the chart, the term "Sample Labelled" means what the classifier thinks is the correct class. We still get some non-storm and Convection Core pixels, since the classifier is not 100% accurate. Row three shows the distribution if we only sample pixels the RDF classifies as Convection Core. Again, we get other classes as well due to inaccuracies in the classifier. This data shows that sampling a Rainy Anvil or Convection Core classified pixel is far more fruitful scientifically than a random pixel.

In order to evaluate improvements in mission return, we utilized our mission simulations using the GWRF datasets. In the simulations, intelligent targeting attempts to preferentially target areas of Convection Core and Rainy Anvil but is limited by the rarity of such pixels, mission energy, and pointing constraints [21]. In Table 13a, row 4 shows the distribution of pixels we would acquire if the classifier used in the simulation was 100% accurate. Row 5 shows the distribution of pixels (true labels) we actually acquire, quantifying how classification inaccuracy reduces the impact on return. However, this mode still dramatically outperforms uninformed (random) targeting. This highlights how preferential targeting is able to skew sampled pixels towards storm-cloud classes.

When looking at the non-tropical dataset, distinguishing between the two classes of Rainy Anvil and convection core is significantly harder. Again, even with the lower classification accuracy, Table 13b shows the expected pixels observed sampling from the Rainy Anvil and Convection Core labelled pixels compared to random sampling. Again, drawing on the mission simulations, the last two rows of Table 13b show dramatically increased yields of Rainy Anvil and Convection Core measurements from intelligent targeting compared to uninformed (random) targeting even with imperfect classification.

| | Clear, Thin Cirrus, and Cirrus | Rainy Anvil | Convection Core |
|---|---|---|---|
| Sample Randomly | 67.0% | 32.0% | 1.0% |
| Sample Labelled as Rainy Anvil | 16.6% | 81.7% | 1.8% |
| Sample Labelled as Convection Core | 12.0% | 64.4% | 23.6% |
| Preferential Targeting (as labelled) | 10.5% | 64.2% | 25.3% |
| Preferential Targeting (true labels) | 23.2% | 69.7% | 7.1% |

**(a) Tropical Data**

| | Clear, Thin Cirrus, and Cirrus | Rainy Anvil | Convection Core |
|---|---|---|---|
| Sample Randomly | 67.0% | 32.0% | 1.0% |
| Sample Labelled as Rainy Anvil | 46.3% | 35.8% | 18.0% |
| Sample Labelled as Convection Core | 76.9% | 10.9% | 12.2% |
| Preferential Targeting (as labelled) | 44.6% | 21.3% | 34.1% |
| Preferential Targeting (true labels) | 76.1% | 14.1% | 9.8% |

**(b) Non-Tropical Data**

**Table 13 Mission return impact of Intelligent Targeting with Classifier**

## XIV. Future Work

We would like to extend our datasets to additional regions beyond the Caribbean and Atlantic coast to make our classifiers more robust. A global dataset could be used to explore the possibility of a universal classifier that would work in any region (and season, and other conditions), however variations in atmospheric phenomena in different regions could make this difficult. Swapping between different regional classifiers in flight would be feasible.

The impact of the expected noise should also be analyzed more clearly. Even though the overall accuracy is not strongly impacted, it is important to know if any cloud types are being disproportionately affected by the noise or if it is balanced.

Future work on the classifiers will expand beyond single pixel classification and take into account surrounding pixels. This should improve the accuracy because storm phenomena are not randomly distributed across the sky, instead they are clustered close together. Upsampling on the storm clouds in each dataset may also improve the overall performance since the datasets are very imbalanced and some of the classifiers that did not have their weights equalized.

It is also important to run the targeting simulation over a dataset that is representative of what SMICES would see in a real flight on future iterations of this work. This would give a better understanding of the improvements that we would expect to see through the targeting algorithms. Additionally, future work can analyze the benefit of tracking a storm across the radar's field of view. This would enable analysis on the morphology of the cloud over its observation time.

Analysis on how the radiometer resolution could impact the performance of the cloud classification is currently unexplored. The targeting algorithm currently assumes the radiometer records data at the GWRF pixel size of 15km

x 15km. Instead the resolution of the radiometer data varies based on the frequency. Understanding impact of combining the different resolutions on the classification accuracy would enable a more accurate simulation of the in flight environment.

Further research also needs to be conducted on the impact that off-nadir analysis has on the data quality for SMICES. Understanding this impact along with how off-nadir the data collected from each algorithm is will allow for a more informed decision between the wide and non-wide variation of the greedy algorithm.

We intend to test these classifiers and targeting algorithms on real data from airborne tests of the SMICES radar.

## XV. Conclusion

We have described an effort to develop a classifier of deep convective storms based on radiometer data. Using a digital twin and K-means clustering we were able to generate labels for data. The results of the classifiers are promising for distinguishing deep convective storms in the tropical dataset. Further work still needs to be done for finer grained storm type discrimination. Additionally, identification of deep convective storms in non tropical data was more challenging.

The results of the targeting trial are very promising for the effectiveness of smart targeting. When comparing the best performing algorithm, greedy path, with the baseline random algorithm, there is a 23.7x increase in the number of convection core pixels analyzed and an almost 2x increase in rainy anvil pixels analyzed. It is important to note that these results are skewed by the dataset used, which contains a much higher proportion of clouds than we would expect to see in the real world. A dataset closer to a real-world scenario would result in a smaller increase in performance.

We also present results indicating that even moderate classification accuracy combined with intelligence instrument targeting are expected to enable significant improvements in mission return.

## XVI. Acknowledgements

The research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

## References

[1] Luo, Z., and Rossow, W. B., "Characterizing Tropical Cirrus Life Cycle, Evolution, and Interaction with Upper-Tropospheric Water Vapor Using Lagrangian Trajectory Analysis of Satellite Observations," *Journal of Climate*, Vol. 17, No. 23, 2004, pp. 4541 – 4563. https://doi.org/10.1175/3222.1.

[2] Stephens, G. L., "Cloud Feedbacks in the Climate System: A Critical Review," *Journal of Climate*, Vol. 18, No. 2, 2005, pp. 237 – 273. https://doi.org/10.1175/JCLI-3243.1.

[3] Bony, S., Colman, R., Kattsov, V., Allan, R., Bretherton, C., Dufresne, J., Hall, A., Hallegatte, S., Holland, M., Ingram, W.,

Randall, D., Soden, B., Tselioudis, G., and Webb, M., "How Well Do We Understand and Evaluate Climate Change Feedback Processes?" *Journal of Climate*, Vol. 19, No. 15, 2006, pp. 3445 – 3482. https://doi.org/10.1175/JCLI3819.1.

[4] King, M. D., Platnick, S., Menzel, W. P., Ackerman, S. A., and Hubanks, P. A., "Spatial and Temporal Distribution of Clouds Observed by MODIS Onboard the Terra and Aqua Satellites," *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 51, No. 7, 2013, pp. 3826–3852. https://doi.org/10.1109/TGRS.2012.2227333.

[5] Taravat, A., Del Fratehttps://www.overleaf.com/project/61cc92e55dd9a96cca44bc69, F., Cornaro, C., and Vergari, S., "Neural Networks and Support Vector Machine Algorithms for Automatic Cloud Classification of Whole-Sky Ground-Based Images," *IEEE Geoscience and Remote Sensing Letters*, Vol. 12, No. 3, 2015, pp. 666–670. https://doi.org/10.1109/LGRS.2014.2356616.

[6] Heinle, A., Macke, A., and Srivastav, A., "Automatic cloud classification of whole sky images," *Atmospheric Measurement Techniques*, Vol. 3, No. 3, 2010, pp. 557–567. https://doi.org/10.5194/amt-3-557-2010, URL https://amt.copernicus.org/articles/3/557/2010/.

[7] Zhuo, W., Cao, Z., and Xiao, Y., "Cloud Classification of Ground-Based Images Using Texture–Structure Features," *Journal of Atmospheric and Oceanic Technology*, Vol. 31, No. 1, 2014, pp. 79 – 92. https://doi.org/10.1175/JTECH-D-13-00048.1, URL https://journals.ametsoc.org/view/journals/atot/31/1/jtech-d-13-00048_1.xml.

[8] Zhang, J., Liu, P., Zhang, F., and Song, Q., "CloudNet: Ground-Based Cloud Classification With Deep Convolutional Neural Network," *Geophysical Research Letters*, Vol. 45, No. 16, 2018, pp. 8665–8672. https://doi.org/https://doi.org/10.1029/2018GL077787.

[9] Li, Q., Zhang, Z., Lu, W., Yang, J., Ma, Y., and Yao, W., "From pixels to patches: a cloud classification method based on a bag of micro-structures," *Atmospheric Measurement Techniques*, Vol. 9, No. 2, 2016, pp. 753–764. https://doi.org/10.5194/amt-9-753-2016, URL https://amt.copernicus.org/articles/9/753/2016/.

[10] Liu, L., Sun, X., Chen, F., Zhao, S., and Gao, T., "Cloud Classification Based on Structure Features of Infrared Images," *Journal of Atmospheric and Oceanic Technology*, Vol. 28, No. 3, 2011, pp. 410 – 417. https://doi.org/10.1175/2010JTECHA1385.1, URL https://journals.ametsoc.org/view/journals/atot/28/3/2010jtecha1385_1.xml.

[11] Desbois, M., Seze, G., and Szejwach, G., "Automatic Classification of Clouds on METEOSAT Imagery: Application to High-Level Clouds," *Journal of Applied Meteorology and Climatology*, Vol. 21, No. 3, 1982, pp. 401 – 412. https://doi.org/10.1175/1520-0450(1982)021<0401:ACOCOM>2.0.CO;2, URL https://journals.ametsoc.org/view/journals/apme/21/3/1520-0450_1982_021_0401_acocom_2_0_co_2.xml.

[12] Seze, G., and Desbois, M., "Cloud Cover Analysis from Satellite Imagery Using Spatial and Temporal Characteristics of the Data," *Journal of Applied Meteorology and Climatology*, Vol. 26, No. 2, 1987, pp. 287 – 303. https://doi.org/10.1175/1520-0450(1987)026<0287:CCAFSI>2.0.CO;2, URL https://journals.ametsoc.org/view/journals/apme/26/2/1520-0450_1987_026_0287_ccafsi_2_0_co_2.xml.

[13] Ambroise, C., Sèze, G., Badran, F., and Thiria, S., "Hierarchical clustering of self-organizing maps for cloud classification," *Neurocomputing*, Vol. 30, No. 1, 2000, pp. 47–52. https://doi.org/10.1016/S0925-2312(99)00141-1, URL https://www.sciencedirect.com/science/article/pii/S0925231299001411.

[14] Candela, A., Swope, J., Chien, S., Su, H., and Tavallali, P., "Dynamic Targeting for Improved Tracking of Storm Features," 2022, pp. 5313–5316. https://doi.org/10.1109/IGARSS46834.2022.9883696, URL https://ai.jpl.nasa.gov/public/documents/papers/IGARSS-2022-Candela-DT.pdf.

[15] Candela, A., Swope, J., and Chien, S., "Dynamic Targeting to Improve Earth Science Missions," *Journal of Aerospace Information Systems (JAIS)*, Vol. 20, No. 11, 2023, pp. 679–689. https://doi.org/10.2514/1.I011233, URL https://doi.org/10.2514/1.I011233.

[16] Rabideau, G., Chien, S., and McLaren, D., "Tractable Goal Selection for Embedded Systems with Oversubscribed Resources," *Journal of Aerospace Computing Information, and Communication (JACIC)*, Vol. 8 (5), 2011, pp. 151–169. https://doi.org/10.2514/1.48919.

[17] Thompson, D., Green, R., Keymeulen, D., Lundeen, S., Mouradi, Y., Nunes, D., Castaño, R., and Chien, S., "Rapid Spectral Cloud Screening Onboard Aircraft and Spacecraft," *Geoscience and Remote Sensing, IEEE Transactions on*, Vol. 52, 2014, pp. 6779–6792. https://doi.org/10.1109/TGRS.2014.2302587.

[18] Suto, H., Kataoka, F., Kikuchi, N., Knuteson, R. O., Butz, A., Haun, M., Buijs, H., Shiomi, K., Imai, H., and Kuze, A., "Thermal and near-infrared sensor for carbon observation Fourier transform spectrometer-2 (TANSO-FTS-2) on the Greenhouse gases Observing SATellite-2 (GOSAT-2) during its first year in orbit," *Atmospheric Measurement Techniques*, Vol. 14, No. 3, 2021, pp. 2013–2039. https://doi.org/10.5194/amt-14-2013-2021, URL https://amt.copernicus.org/articles/14/2013/2021/.

[19] Hasnain, Z., Mason, J., Swope, J., Vander Hook, J., and Chien, S., "Agile Spacecraft Imaging Algorithm Comparison for Earth Science," 2021. URL https://ai.jpl.nasa.gov/public/papers/Hasnain-IWPSS2021-paper-13.pdf.

[20] Candela, A., Swope, J., and Chien, S., "Dynamic Targeting for Cloud Avoidance to Improve Science of Space Missions," *16th Symposium on Advanced Space Technologies in Robotics and Automation*, 2022. URL https://ai.jpl.nasa.gov/public/documents/papers/Candela-DT-ASTRA-2022.pdf.

[21] Swope, J., Chien, S., Bosch-Lluis, X., Yue, Q., Tavallali, P., Ogut, M., Ramos, I., Kangaslahti, P., Deal, W., and Cooke, C., "Using Intelligent Targeting to increase the science return of a Smart Ice Storm Hunting Radar," 2021. URL https://ai.jpl.nasa.gov/public/documents/papers/Swope-SMICES-targeting-IWPSS-2021.pdf.

[22] Swope, J., Mirza, F., Dunkel, E., Candela, A., Chien, S., Holloway, A., Russell, D., Sauvageau, J., Sheldon, D., and Fernandez, M., "Benchmarking Space Mission Applications on the Snapdragon Processor Onboard the ISS," *Journal of Aerospace Information Systems*, Vol. 20, No. 12, 2023, pp. 807–816. https://doi.org/10.2514/1.I011217, URL https://arc.aiaa.org/doi/10.2514/1.I011217.

[23] Swope, J., Mirza, F., Dunkel, E., Towfic, Z., Russell, D., Sauvageau, J., Sheldon, D., Chien, S., Fernandez, M., and Knox, C., "Benchmarking Remote Sensing Image Processing and Analysis on the Snapdragon Processor Onboard the International

Space Station," 2022. https://doi.org/10.2514/1.I011217, URL https://ai.jpl.nasa.gov/public/documents/papers/IGARSS2022-Onboard-Not-DL-IGARSS2022-Camera.pdf.

[24] "Rad750," https://en.wikipedia.org/wiki/RAD750, retrieved 12 October 2022.

[25] Skamarock, W.C., Klemp, J., Dudhia, J., Gill, D.O., Liu, Z., Berner, J., Wang, W., Powers, G., Duda, M., Barker, D., and X.-Y., H., "A Description of the Advanced Research WRF Version 4," *NCAR Tech. Note NCAR/TN-556+STR*, 2019, p. 145 pp. https://doi.org/10.5065/1dfh-6p97.

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

[27] Chollet, F., et al., "Keras," https://keras.io, 2015.

[28] Abadi, M., and et. al., "TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems," , 2015. URL https://www.tensorflow.org/, software available from tensorflow.org.

[29] Kingma, D. P., and Ba, J., "Adam: A Method for Stochastic Optimization," , 2017. https://doi.org/10.48550/arXiv.1412.6980.