Lab 2 - Cloud Data, Stat 214, Spring 2025

March 21, 2025

1. Introduction

Accurate cloud detection in the Arctic is critical for climate modeling, particularly because polar regions are expected to experience the most pronounced effects of climate change. However, distinguishing clouds from snow- and ice-covered surfaces remains a challenge due to their similar spectral signatures in traditional visible and thermal imagery. To address this, Shi et al. (2008) developed novel Arctic cloud detection algorithms based on multi-angle satellite data from NASA's MISR instrument. Their approach uses engineered features such as CORR (angular correlation), SD (nadir spatial smoothness), and NDAI (normalized difference angular index) to effectively separate cloud-free surfaces from cloudy regions. Building on this work, our lab leverages both these engineered features and deep learning-based latent features from an autoencoder trained on MISR data to build robust cloud classifiers.

Given the scarcity of expert-labeled images in our dataset, we employed a transfer learning strategy that integrates spatial and unsupervised features for supervised classification. Through exploratory data analysis (EDA), we identified key patterns in cloud distribution and selected top-performing features. To assess generalization, we evaluated the model's stability under simulated sensor noise—using both additive and multiplicative perturbations—and observed minimal change in predictions. We further applied the model to unlabeled images and found that it consistently predicted spatially coherent cloud regions. These results indicate strong generalization to unseen data and robustness to small input perturbations, suggesting that our classifier is well-suited for real-world deployment where labels are scarce, as long as the data distribution remains similar to the training set.

2. EDA

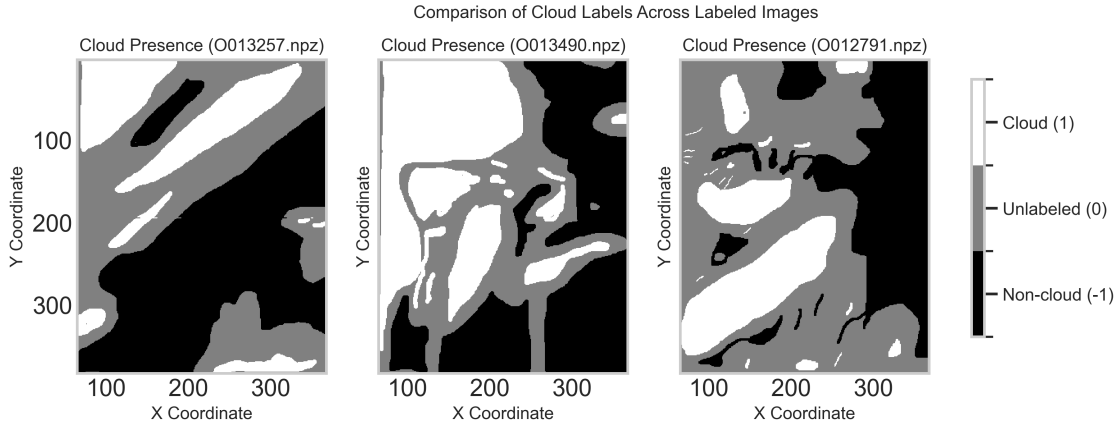
2.1 Visualizing Expert Labels on Satellite Images

The visualization above illustrates cloud label distributions across the three labeled satellite images: O013257.npz, O013490.npz, and O012791.npz. In this grayscale encoding, cloud pixels are shown in black (label = 1), non-cloud pixels in white (label = -1), and unlabeled regions in gray (label = 0). This representation facilitates visual comparison of spatial labeling across different scenes using consistent X and Y coordinates.

The first image (O013257.npz, left panel) exhibits a relatively large presence of cloud cover, particularly along diagonal bands from the bottom left to the top right. The cloud and non-cloud regions are distinctly separated, with relatively fewer ambiguous or unlabeled areas. In contrast, the second image (O013490.npz, middle panel) presents a more fragmented distribution, where cloud and non-cloud pixels are interleaved in irregular patterns. This image features more complex cloud boundaries, likely increasing classification difficulty due to the lack of large, continuous cloud regions. The third image (O012791.npz, right panel) contains smaller, more scattered cloud formations, intermixed with significant non-cloud and unlabeled areas, especially in the lower half of the image. The scattered cloud patches and finer structural details suggest a more challenging detection scenario.

The variation in cloud density, spatial arrangement, and label coverage across the three images high-

lights the inherent complexity of real-world atmospheric conditions. These differences underscore the need for robust and adaptable cloud detection methods capable of generalizing across diverse spatial and textural patterns. This aligns with findings from Shi et al. (2008), who emphasize that accurate cloud detection in satellite imagery—particularly over complex surfaces such as ice, snow, and varied terrains—requires careful feature design and spatially aware validation. As their work on Arctic cloud detection demonstrates, combining expert-informed features with adaptive, region-sensitive modeling approaches can significantly improve detection accuracy. In our case, incorporating both dense and fragmented cloud examples during training ensures broader generalizability, preparing the model to perform reliably across images with varying cloud formations and geographic characteristics.

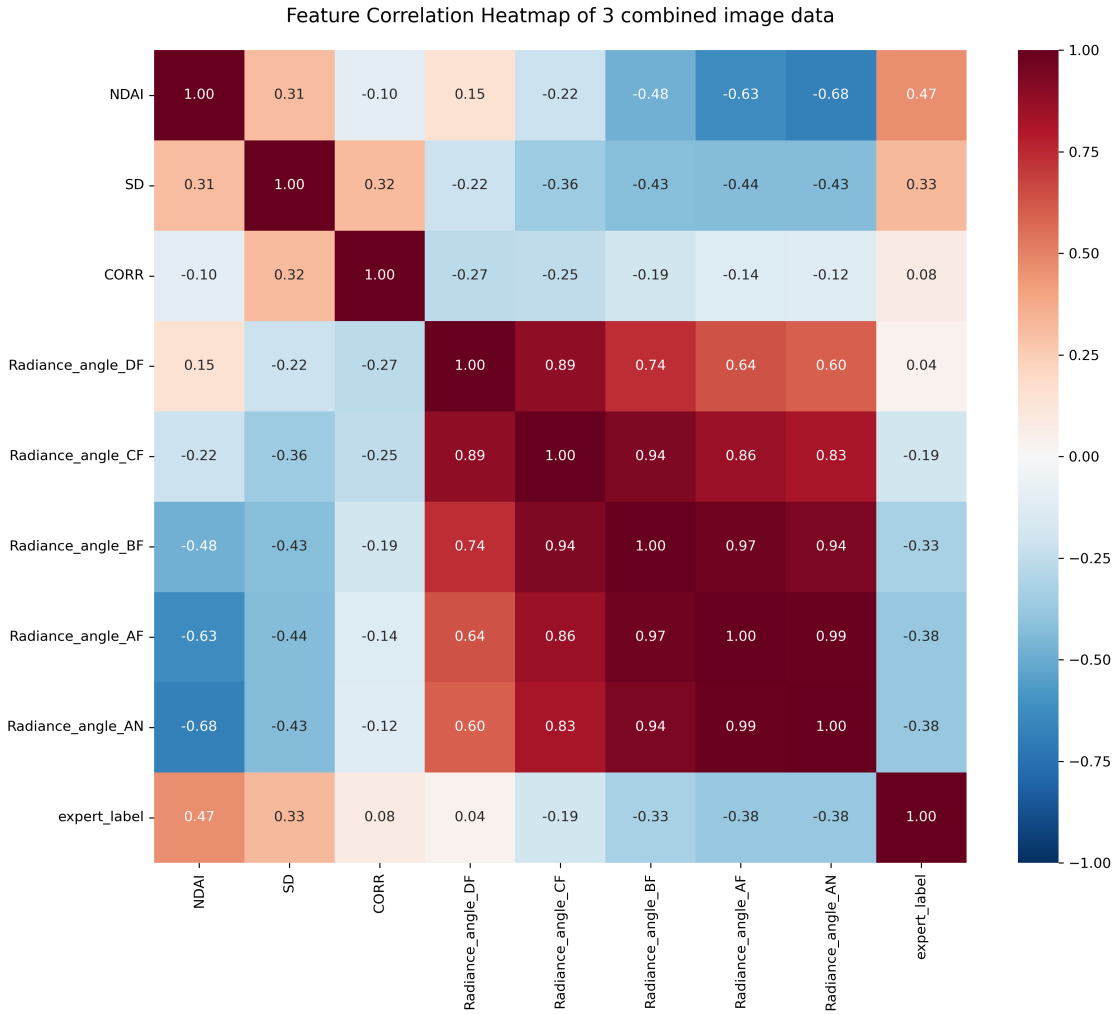


2.2 Radiance Feature Analysis Across Viewing Angles

The feature correlation heatmap reveals high positive correlations among different radiance angles (e.g., Radiance_angle_CF and Radiance_angle_BF have a correlation of 0.94), suggesting that radiances from different angles follow similar patterns, likely due to shared atmospheric and surface properties. The highest correlations occur between adjacent angles, indicating that slight changes in angle do not significantly alter radiance measurements. NDAI shows a moderate positive correlation with expert labels (0.47), suggesting its relevance for cloud detection, while SD (0.33 correlation with expert labels) indicates that pixel intensity variability may help distinguish clouds from non-cloud regions.

However, individual radiance angles, such as Radiance_angle_CF (-0.19) and Radiance_angle_BF (-0.33), exhibit weak correlations with expert labels, implying they are not strong standalone predictors of cloud presence. Similarly, the CORR feature has a very weak correlation (0.08) with expert labels, suggesting limited usefulness in cloud classification. NDAI also exhibits negative correlations with Radiance_angle_AF (-0.63) and Radiance_angle_AN (-0.68), indicating that different angles capture cloud presence in distinct ways.

The high intercorrelations among radiance angles suggest redundancy, meaning dimensionality reduction techniques such as PCA could help minimize feature overlap. While NDAI and SD appear to be useful features for cloud classification, radiance angles alone are insufficient predictors, reinforcing the need for a robust cloud detection model that integrates multiple features rather than relying solely on radiance measurements from individual angles.

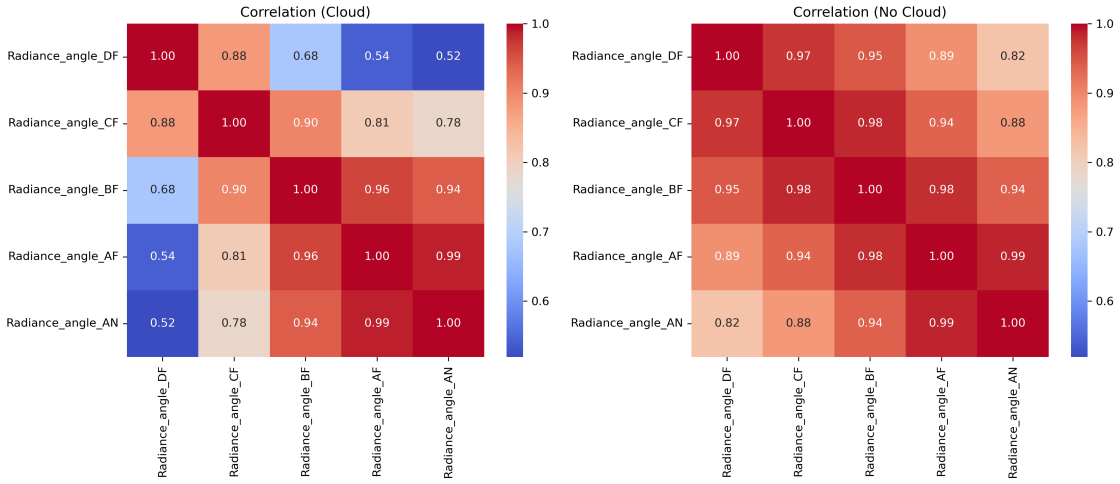


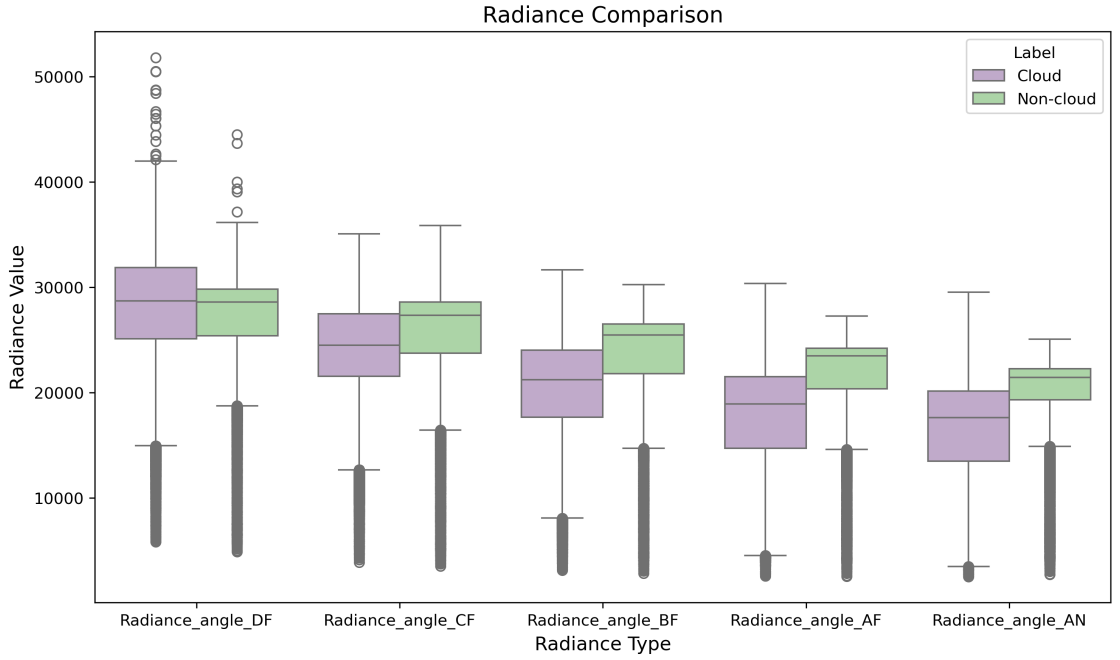
Under no-cloud conditions, radiance angles exhibit strong correlations, mostly above 0.94, indicating a consistent radiance pattern across viewing angles. This suggests that when no clouds are present, radiance values remain stable regardless of the observation angle. However, cloudy conditions lead to lower and more variable correlations, suggesting disrupted radiance patterns due to cloud scattering and absorption. The heatmap further reinforces this, showing that radiance correlations remain high in clear conditions but weaken significantly in the presence of clouds, demonstrating how clouds disrupt radiance consistency across viewing angles.

Among all angles, Radiance_angle_DF experiences the most substantial correlation drop in cloudy conditions, particularly with Radiance_angle_AF (0.54 vs. 0.89 in no-cloud conditions), indicating its higher sensitivity to cloud presence. While some angle pairs, such as AF–AN and BF–AF, maintain relatively high correlations, their subtle reductions under clouds could serve as key indicators for cloud detection.

The boxplots show that non-cloud pixels consistently exhibit higher median radiance across all angles, with the difference becoming more pronounced at larger observation angles (AF, AN), suggesting these angles are more effective for cloud discrimination. Cloud-covered regions display greater variability and more outliers, reflecting the heterogeneous nature of cloud structures, while non-cloud samples exhibit a more uniform and predictable radiative behavior.

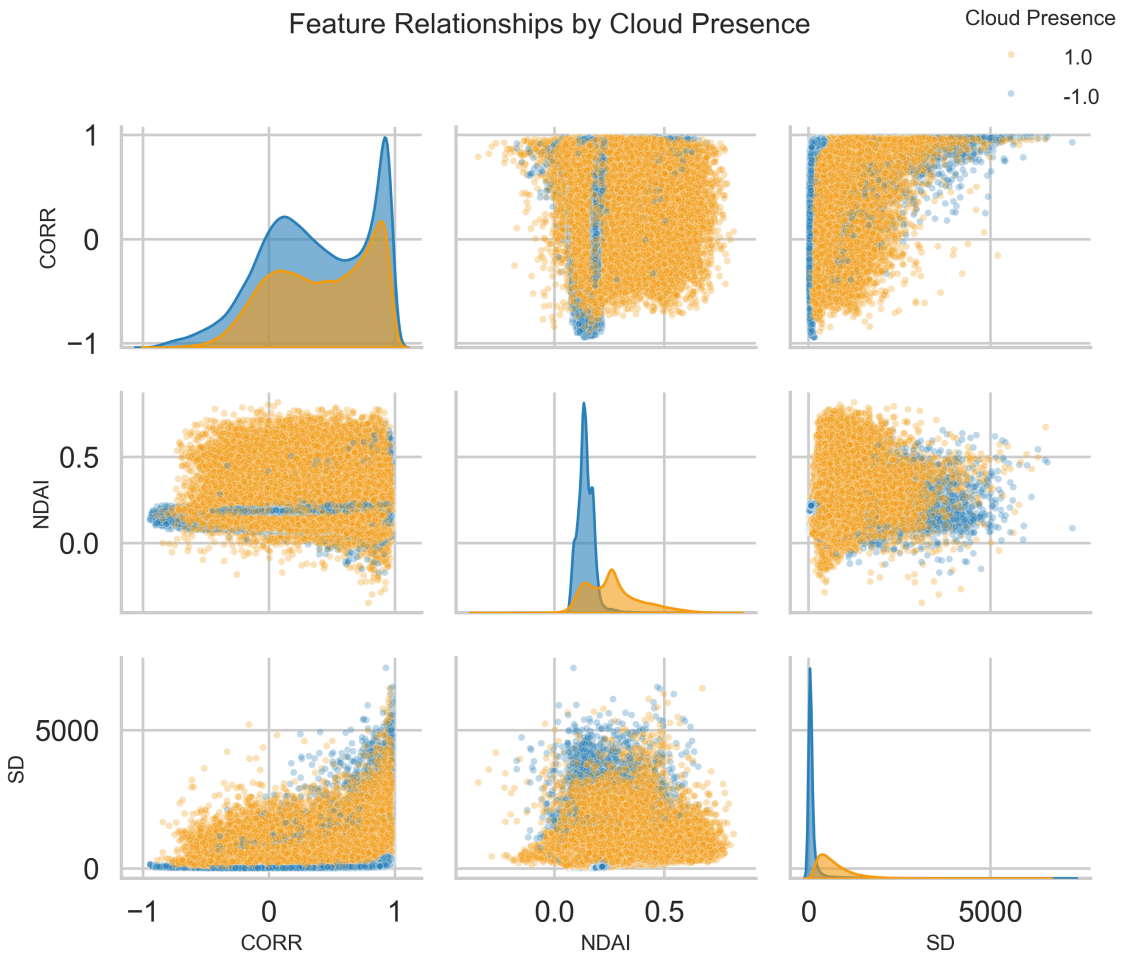
Additionally, cloud regions tend to have a wider interquartile range and a higher density of outliers, particularly at Radiance_angle_DF and Radiance_angle_CF, suggesting greater unpredictability in cloud radiance. The presence of multiple extreme outliers in cloud regions may indicate high variability in cloud reflectance, possibly due to differences in cloud density, thickness, or multi-layer formations. These findings emphasize the need for adaptive thresholding techniques in cloud detection models to account for varying radiance behaviors.





The analysis of CORR, NDAI, and SD reveals clear differences in their effectiveness for distinguishing cloud and non-cloud regions. NDAI and SD emerge as the most informative features, with NDAI values being generally higher for clouds, indicating its strong ability to separate cloud presence. SD, which measures variability, is also higher in cloud regions, likely due to increased fluctuations in reflectance, as seen in the scatterplots. The distribution of SD has a long tail, meaning cloud pixels exhibit greater radiance variability, making SD a useful indicator of cloud presence. In contrast,

CORR does not provide a strong distinction between cloud and non-cloud regions, as its distribution shows significant overlap between the two classes. These findings suggest that a combination of NDAI and SD is the most effective approach for cloud classification, with NDAI showing the strongest separation and SD providing complementary variability information to enhance detection accuracy.



2.3 Data Splitting Strategy

We implemented a two-level spatial data splitting strategy to support generalization in cloud classification. First, we split the data at the image level: two expertly labeled images were used for training and validation, and the third image was held out as an independent test set to evaluate performance on unseen spatial regions. Within the training and validation set, we further divided the data into four spatial quadrants based on the median x and y coordinates, ensuring that training and validation data came from distinct parts of the image. Unlabeled pixels were excluded from all subsets.

To ensure a fair evaluation, we iterated through different train-validation-test combinations using three available expert-labeled images (O013257.npz, O013490.npz, O012791.npz), where each image contains an additional column of expert annotations. In each iteration, two images were assigned for training and validation, while the third was held out for testing. After this image-level split,

we applied quadrant-based partitioning within the training and validation set, ensuring spatially disjoint data partitions to minimize spatial leakage and promote robust generalization.

This approach is motivated by the challenges outlined in Shi et al. (2008), who highlight the difficulty of distinguishing clouds from ice- and snow-covered surfaces due to their similar radiative properties. They stress the importance of evaluating models on spatially disjoint data to ensure robust generalization, especially for operational cloud detection tasks across large-scale satellite imagery. Their findings show that models trained and tested on overlapping regions tend to overfit, whereas separating data by space and time better reflects real-world deployment scenarios. Our image-level and quadrant-based splitting mirrors this principle by preventing spatial leakage and promoting more realistic model evaluation.

2.4 Data Cleaning and Handling Imperfections

As with many real-world datasets, the satellite images contained imperfections—most notably, a subset of pixels without expert-provided labels. We initially considered converting these unlabeled pixels into pseudo-labels (e.g., 0 for non-cloud, 1 for cloud) and applying unsupervised learning techniques. This could have expanded the training set and potentially improved the model's ability to generalize, particularly in ambiguous or boundary regions where cloud classification is more challenging. However, upon closer inspection, we found no missing feature values, and given the uncertainty in assigning reliable pseudo-labels and the risk of introducing label noise, we ultimately decided to exclude these pixels from our analysis rather than impute or infer their class.

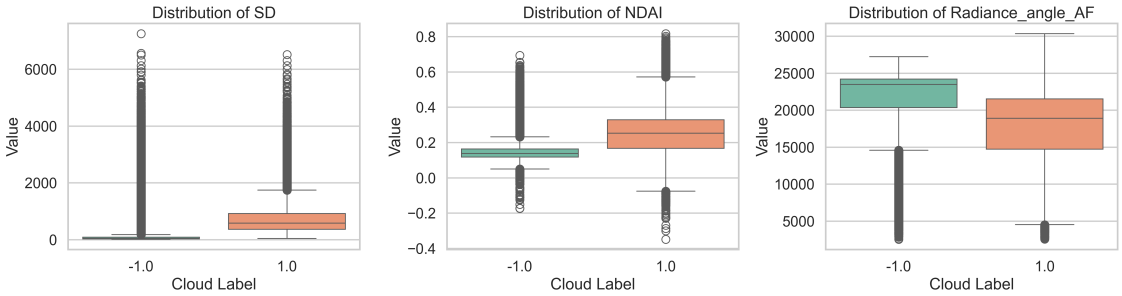
3. Feature Engineering

3.1 Identifying Informative Features

To identify the most informative features for cloud classification, we analyzed their statistical significance using three key metrics: Point-Biserial Correlation Coefficient (PBCC) to measure the strength of association between each feature and the binary cloud label, Mutual Information (MI) to capture both linear and non-linear dependencies, and the ANOVA F-test to assess the degree of separability between cloud and non-cloud regions. The statistical significance of each feature was further evaluated using p-values. The results indicate that SD (Standard Deviation), NDAI (Normalized Difference Airborne Index), and Radiance_angle_AF are the most relevant features, consistently ranking high across multiple statistical measures. Specifically, NDAI exhibited the highest correlation with cloud presence (PBCC = 0.4668) and the highest F-score (50,962.25), making it a strong predictor of cloud regions. SD showed the highest MI score (0.3509), indicating a strong relationship with cloud classification, while its high F-score (42,512.24) further supports its predictive power. Although Radiance_angle_AF had a negative correlation (PBCC = -0.3791), it maintained a meaningful MI score (0.1529) and an F-score of 29,271.49, suggesting that variations in radiance angles contribute significantly to cloud detection.

The statistical significance of these findings is supported by p-values close to zero for all features, confirming their relevance, except for Radiance_angle_DF, which, while small, has a nonzero p-value (1.62e-254). The feature distributions further validate these results, as SD and NDAI show clear separability between cloud and non-cloud labels, reinforcing their high statistical rankings. Radiance_angle_AF also demonstrates distinct variations in median values across cloud and non-cloud pixels, supporting its significance despite its negative correlation. Other features, such as Radiance_angle_AN and Radiance_angle_BF, exhibit moderate importance, while CORR and

Radiance_angle_CF have the lowest MI and F-scores, indicating a weaker contribution to classification performance. These results highlight SD, NDAI, and Radiance_angle_AF as primary predictors for cloud classification, with potential for further refinement through advanced feature selection techniques to optimize model performance.



3.2 Designing New Features Using Spatial Context

To enhance the predictive power of cloud classification models, we engineered a comprehensive set of spatially-aware features based on the information contained in local image patches surrounding each pixel. Specifically, we computed summary statistics—including the mean, standard deviation, minimum, and maximum—for each of the eight primary expert-derived channels (NDAI, SD, CORR, DF, CF, BF, AF, and AN) across a 9×9 patch centered at each labeled pixel. These new patch-based features aim to capture local spatial patterns and texture variations in cloud structure that may not be evident from individual pixel-level values alone. By integrating this localized context, we can better account for smoothness, variability, and spatial gradients, which are often indicative of cloud presence. The resulting dataset comprises 32 new features in addition to the original coordinates and expert label, providing a richer representation of pixel neighborhoods that facilitates more accurate classification. This approach leverages domain knowledge while also incorporating data-driven spatial context, making it a robust and interpretable feature engineering strategy.

To generate these features, the labeled satellite image data was first reshaped onto a unified spatial grid using global minimum and maximum x and y coordinates, ensuring spatial consistency across all images. Feature values across the eight channels were globally normalized to zero mean and unit variance. Each image was then padded using reflection to allow for patch extraction near the edges. For every labeled pixel, a 9×9 patch was extracted from each normalized channel, and the four summary statistics were computed over that patch. These values were stored alongside the corresponding pixel coordinates and expert labels, resulting in a structured dataset enriched with meaningful spatial context for downstream modeling.

3.3 Transfer Learning

Transfer learning is a machine learning approach that enables a model to transfer knowledge gained from one task or dataset to another, typically related, task. This method is particularly advantageous when labeled data is scarce, as it allows a model to leverage patterns learned from abundant unlabeled data to improve performance on a smaller, supervised task. In this project, transfer learning was implemented by first pre-training an autoencoder on a large set of unlabeled satellite image patches and subsequently fine-tuning it on three labeled training images. Although fine-tuning was conducted on labeled data, expert annotations were not used in this phase. Instead, the autoencoder continued to learn in an unsupervised fashion by minimizing reconstruction error, allowing it

to capture structural features in cloud formations, brightness variations, and terrain patterns.

The autoencoder was designed to process 9×9 patches extracted from multi-spectral satellite images, each patch containing 8 spectral channels. The model encodes these patches into a compressed latent space and reconstructs the original input from this low-dimensional representation. Two architectures were evaluated during development: a convolutional autoencoder (ConvAE) and a fully connected autoencoder (FC-AE). The convolutional variant incorporated several convolutional and transposed convolutional layers to exploit local spatial structures. However, due to the small patch size and aggressive downsampling, the ConvAE struggled to preserve sufficient spatial detail, resulting in poorer reconstruction performance. As a result, we adopted the fully connected architecture from the starter code as our base model but introduced several key modifications to enhance its capacity.

The fully connected autoencoder was deepened by adding additional hidden layers to both the encoder and decoder. The encoder sequentially transformed the flattened input through progressively smaller layers, compressing it into a 50-dimensional latent representation. The decoder mirrored this structure to reconstruct the original input. Each layer was followed by ReLU activations, and Batch Normalization was applied between layers to stabilize training and mitigate internal covariate shifts. The final model captured non-linear relationships within the patches more effectively than the original two-layer design.

Hyperparameter tuning was an integral part of optimizing the autoencoder's performance. We experimented with embedding sizes ranging from 8 to 64 and ultimately selected 50, which provided a good balance between compression and reconstruction fidelity. Learning rates between 10^3 and 10 were tested, and while 0.001 accelerated training, a learning rate of 0.0001 yielded better generalization and more stable convergence. To evaluate the impact of batch size, we trained models using batch sizes of 512, 1028, and 8192. Among these, a batch size of 1028 achieved the lowest validation loss while maintaining efficient training dynamics. The model was trained for up to 100 epochs, with validation loss monitored to select the best checkpoint.

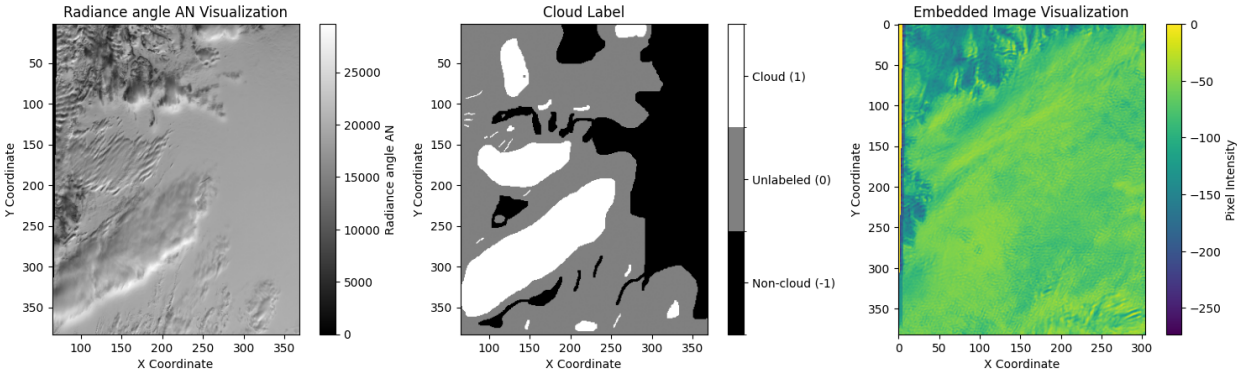
Once trained, the autoencoder served as a feature extractor for the downstream cloud classification task. The 50-dimensional latent vectors produced by the encoder were included as part of the input to the classification models. Although these autoencoder-derived embeddings did not surpass the hand-crafted patch-based features in terms of raw importance, they offered additional abstract representations of cloud texture and atmospheric conditions that contributed to overall model robustness. In particular, the embeddings helped improve generalization by capturing high-level spatial regularities that may not be easily encoded by basic statistical summaries.

To assess the impact of hyperparameter tuning on the quality of learned representations, we visualized the autoencoder embeddings for the labeled image O012791.npz after training with various architectural and training configurations. In the first visualization, we plotted the 50-dimensional latent vectors obtained from the tuned autoencoder using t-SNE projection. The resulting plot revealed partial but not fully distinct separation between cloudy and non-cloudy pixels, suggesting that while the model had learned some discriminative structure, there remained overlap in the feature space. This initial result indicated that the model was beginning to encode meaningful spatial patterns, but further refinement was needed to enhance class separation.

In the second visualization, we introduced additional depth into the autoencoder architecture by increasing the number of hidden layers. This adjustment led to sharper separation between cloud and non-cloud regions in the embedding space. The latent representations exhibited clearer clustering and reduced overlap, demonstrating that a deeper architecture allowed the model to better capture

complex spatial and spectral variations in the image. Among the eight input channels, the AN (nadir-view) angle continued to show strong influence, helping anchor the embedding structure. This observation aligns with Yu et al. (2008), who noted that nadir-view radiance from the AN camera provides minimal angular distortion and is especially useful for distinguishing cloud-covered pixels over reflective surfaces like snow or ice.

The third and final visualization corresponds to the autoencoder trained with the final optimized hyperparameters, including an embedding size of 50, a batch size of 1028, and a learning rate of 0.0001. The resulting embedding showed the clearest and most well-defined separation between cloud and non-cloud classes, validating the effectiveness of our tuning strategy. The tight clustering of latent features by label class suggests that the model successfully internalized discriminative features despite the absence of supervision during training. Once again, the inclusion of radiance data from the AN angle played a key role in driving this separation, reinforcing its importance as highlighted in Shi et al. (2008). These results confirm that with careful architectural design and tuning, the autoencoder can learn robust, transferable representations that support downstream classification even when labeled data is limited.



In summary, we began with a convolutional autoencoder architecture but ultimately adopted a deeper fully connected model after observing improved reconstruction accuracy and embedding quality. Through extensive experimentation with architectural depth, embedding size, learning rate, and batch size, we developed an autoencoder that not only minimized reconstruction error but also produced discriminative, transferable representations of cloud structures. This transfer learning approach significantly enhanced the model's ability to generalize across spatial conditions, supporting more accurate downstream classification in scenarios with limited labeled data. The autoencoder achieved a minimum validation MSE loss of 0.040945.

4. Predictive Modeling

4.1 Data Splitting and Experimental Setup

In this project, we developed and compared several supervised classifiers to predict the presence of clouds in satellite imagery. The dataset originated from three expertly annotated satellite images, where domain experts provided ground truth labels identifying cloudy vs. non-cloudy pixels. These labeled datasets served as the foundation for building machine learning models capable of performing accurate pixel-wise classification.

To enrich the feature space and improve model performance, we incorporated a diverse set of features from multiple sources:

(1) Original expert features (11 features) such as NDAI, SD, CORR, and other hand-engineered statistical descriptors.

(2) New patch-based statistical features (32 features) derived from each pixel's 9×9 spatial neighborhood, capturing mean, standard deviation, minimum, and maximum statistics across eight spectral channels (NDAI, SD, CORR, DF, CF, BF, AF, AN). These features aimed to leverage local spatial structure, which is known to be highly informative for cloud detection.

(3) Autoencoder embeddings (50 features) derived from Part 2 of the project, where a deep autoencoder was trained to learn compressed, latent representations of the image patches. These embeddings abstract complex nonlinear patterns in the data and provide a dense, information-rich feature space for classification.

Altogether, the final feature space used for predictive modeling consisted of 93 features per pixel (11 + 32 + 50), representing a rich fusion of expert knowledge, spatial texture statistics, and deep learned representations.

To ensure a rigorous and generalizable evaluation of our models, we implemented a two-tiered spatial data splitting strategy.

(1) Image-Level Splitting Using CSV Files

The full dataset was divided into two separate CSV files, each derived from the combined feature matrices (93 features) described above:

final_2_images.csv: Included all pixels from two of the expert-labeled images, serving as the basis for training and validation. final_1_images.csv: Contained pixels from the third expert-labeled image, set aside as an independent test set, representing completely unseen spatial data. This mimics a real-world application scenario, where a trained model must generalize to new regions. These CSV files were generated only after the full feature pipeline was complete, ensuring that all 93 features—including autoencoder embeddings and patch-based features—were included in both the training/validation and test datasets.

(2) Intra-Image Spatial Splitting Using Quadrants

Within the final_2_images.csv data, we implemented a spatial quadrant-based split to avoid information leakage and evaluate model generalizability across different spatial zones. We calculated the median x and y coordinates across all pixels in the file and assigned each pixel to one of four quadrants:

- Q1: Top-left
- Q2: Top-right
- Q3: Bottom-left
- Q4: Bottom-right

This spatial disjointing of training and validation samples ensured that model performance reflected spatial generalization, not overfitting to local image structures.

After removing unlabeled pixels (where expert_label == 0), we used this quadrant structure for GroupKFold cross-validation and to split a training set and a validation set while keeping quadrant integrity intact.

The test set (final_1_images.csv) remained fully independent, consisting of all labeled pixels from the third image.

4.2 Classifier Development, Evaluation, and Model Comparison

In this analysis, we developed and evaluated three distinct classification models—Random Forest, K-Nearest Neighbors (KNN), and LightGBM—to identify the presence of clouds in satellite image data. Each classifier was rigorously optimized through grid-based hyperparameter tuning using GroupKFold cross-validation, ensuring that spatial coherence was preserved by assigning validation splits based on distinct quadrants in the data. Our modeling was informed by an expanded feature space that incorporated 32 engineered patch-based features, 50 autoencoder-derived embeddings from Part 2 of the lab, and 11 original pixel-level features, resulting in a rich and high-dimensional dataset for training.

(1) Random Forest Classifier

The first classifier implemented was Random Forest, a powerful ensemble method that builds multiple decision trees and aggregates their predictions to enhance accuracy and reduce overfitting. We conducted extensive grid-based hyperparameter tuning, exploring variations in the number of trees (n_estimators = 100, 200), tree depth (max_depth = 10, 20), and complexity-controlling parameters (min_samples_split = 2, 5 and min_samples_leaf = 1, 2).

After GroupKFold cross-validation, the best-performing configuration consisted of 200 trees, a maximum depth of 10, a minimum of 2 samples required for a split, and a minimum of 2 samples per leaf node. This model delivered the highest accuracy (83%) and an impressive ROC-AUC score of 0.94 on the held-out test set. The classification report showed a well-balanced trade-off between precision and recall, particularly for cloud-present pixels (label 1), making Random Forest the best-performing model in our study.

As a non-parametric ensemble method, Random Forest makes no strict assumptions about feature distributions or class separability. Instead, it relies on recursive binary splits to capture patterns in the data while reducing variance through tree aggregation. Given our high-dimensional and structured dataset, these assumptions are well-suited for the problem. Feature importance analysis confirmed that the model relied on meaningful predictors, while permutation importance tests showed that shuffling key features significantly reduced accuracy—validating that the model's decisions were driven by real patterns rather than noise. Additionally, the absence of overfitting in cross-validation suggests that the assumption of uncorrelated trees holds, reinforcing the model's reliability.

(2) K-Nearest Neighbors (KNN) Classifier

The second classifier we evaluated was K-Nearest Neighbors (KNN), a simple yet intuitive non-parametric method that predicts labels based on the majority vote of the closest neighboring samples in feature space. We tested different values for n_neighbors (ranging from 1 to 4) and found that 4 neighbors provided the best balance between bias and variance. On the test set, the KNN model achieved an accuracy of 82% and an ROC-AUC score of 0.84. While it performed reasonably well, its recall for cloud-present pixels was slightly lower than that of Random Forest, suggesting limitations in detecting subtle spatial and spectral patterns associated with cloud presence.

KNN operates under the assumption that similar instances in feature space should have similar

labels, relying on distance metrics (e.g., Euclidean distance) to define neighborhood relationships. However, in high-dimensional spaces, these distances become less meaningful—an issue known as the curse of dimensionality. Although we standardized the features to mitigate scale-related distortions, the lower recall and ROC-AUC suggest that KNN struggled with the complexity of spatially and spectrally heterogeneous data. This indicates that its fundamental assumption of local similarity was only partially valid in this context, making it less effective for cloud classification compared to ensemble-based models.

(3) LightGBM Classifier

The third classifier evaluated was LightGBM, a gradient boosting framework optimized for speed and efficiency in high-dimensional spaces. LightGBM is particularly well-suited for large datasets like ours and provides a robust framework for modeling non-linear relationships. We tuned a comprehensive grid of hyperparameters, including the number of boosting iterations (n_estimators of 100 and 200), learning rates (0.01 and 0.1), maximum tree depths (5 and 7), number of leaves per tree (31 and 63), and the minimum number of samples required in a leaf node (min_child_samples set to 20 and 50). The best configuration included 200 estimators, a learning rate of 0.1, a tree depth of 5, 63 leaves per tree, and a minimum child sample size of 20. On the test set, the LightGBM model achieved an accuracy of 82% and an ROC-AUC score of 0.93. Although its performance was quite comparable to Random Forest, the slightly lower recall on cloud-positive instances made it the second-best model overall.

Like other gradient boosting methods, LightGBM assumes that complex relationships can be effectively modeled through an additive series of decision trees. It does not require linearity or independence between features and is robust to multicollinearity and missing data. These assumptions align well with our dataset, which contains nonlinear spatial patterns and high-dimensional embeddings. The model's strong performance across multiple metrics suggests that these assumptions were largely met in this context.

Among all three classifiers, Random Forest emerged as the most effective and reliable model, outperforming both KNN and LightGBM across multiple evaluation metrics. With an ROC-AUC score of 0.94, it demonstrated strong discriminative power while maintaining balanced precision and recall for both cloud and non-cloud labels.

A key advantage of Random Forest was its interpretability, which allowed us to analyze feature importance and gain insights into the most influential predictors for cloud classification. LightGBM, while nearly as effective, exhibited slightly lower recall for cloud-positive instances and relied more heavily on boosting optimization. KNN, despite its conceptual simplicity and computational efficiency, struggled in this high-dimensional and spatially heterogeneous dataset, as it lacked the ability to model complex decision boundaries effectively.

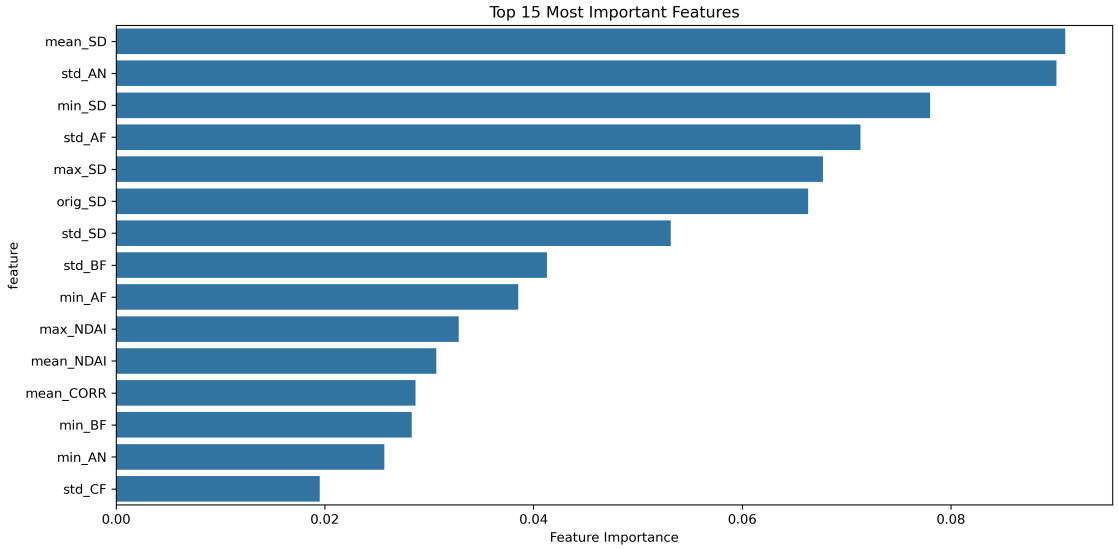
Overall, this comparative modeling exercise highlights the advantages of ensemble-based decision tree models in remote sensing tasks with complex, structured input features. Random Forest delivered top-tier performance while also providing interpretable outputs, making it the most suitable model for cloud classification in this study.

4.3 Evaluation of Random Forest Classifier

To optimize hyperparameters for the Random Forest classifier, we employed Optuna, a flexible and efficient hyperparameter tuning framework that uses sequential trial-based optimization. For

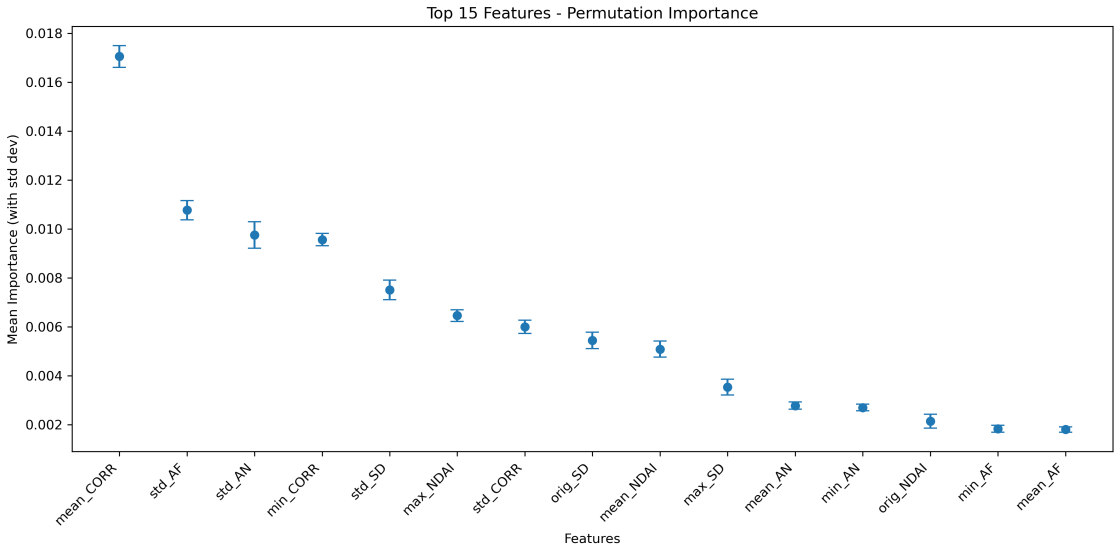
each parameter—such as the number of estimators, maximum tree depth, and minimum samples per split—we defined a lower and upper bound. Optuna then performed a specified number of trials, randomly sampling parameter combinations and evaluating each using cross-validation. The average accuracy across folds was used to identify the best-performing parameter set. In this project, Optuna was applied solely to the Random Forest model, while other classifiers were tuned manually or used default settings. After determining that Random Forest achieved the highest performance, we conducted a comprehensive post-hoc analysis to evaluate its interpretability and robustness, including assessments of feature importance, confidence calibration, misclassification patterns, and spatial distribution of errors. The following sections present these findings in detail, as visualized in Figures 1–5.

(1) Feature Importance via Mean Decrease in Impurity (MDI)



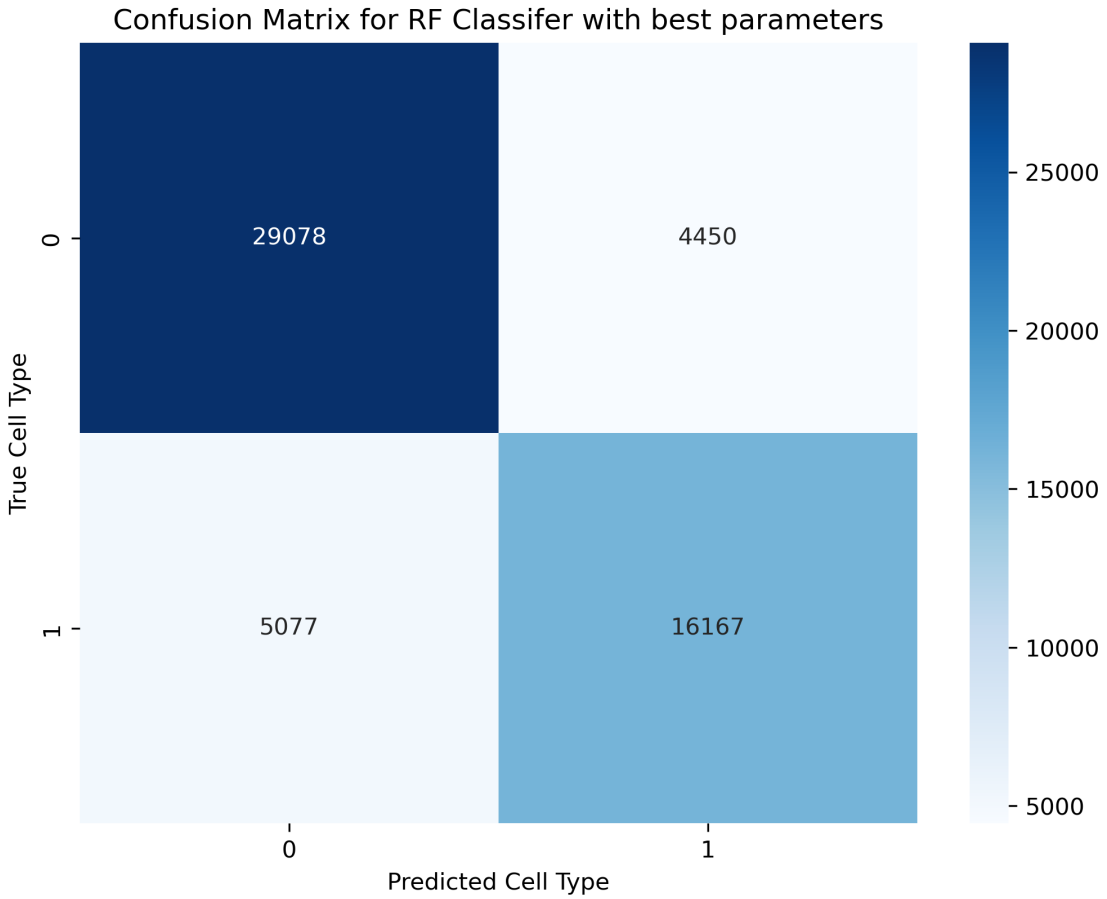
The first diagnostic plot evaluates feature importance based on the Random Forest's internal measure of impurity reduction. As shown in Figure 1, the top contributing features include mean_SD, std_AN, min_SD, std_AF, and max_SD, among others. These features are primarily derived from patch-based statistical aggregates such as means and standard deviations, especially for SD, AF, and AN channels. Interestingly, none of the 50 autoencoder-derived embedding features ranked among the top 15 features. This result suggests that the engineered spatial features—especially those capturing local variation (e.g., std_ and min_)—were more informative for classification than the compressed representations generated via the autoencoder. This supports the effectiveness of our expert-informed patch-based feature engineering over the abstract latent representations.

(2) Permutation Importance Analysis



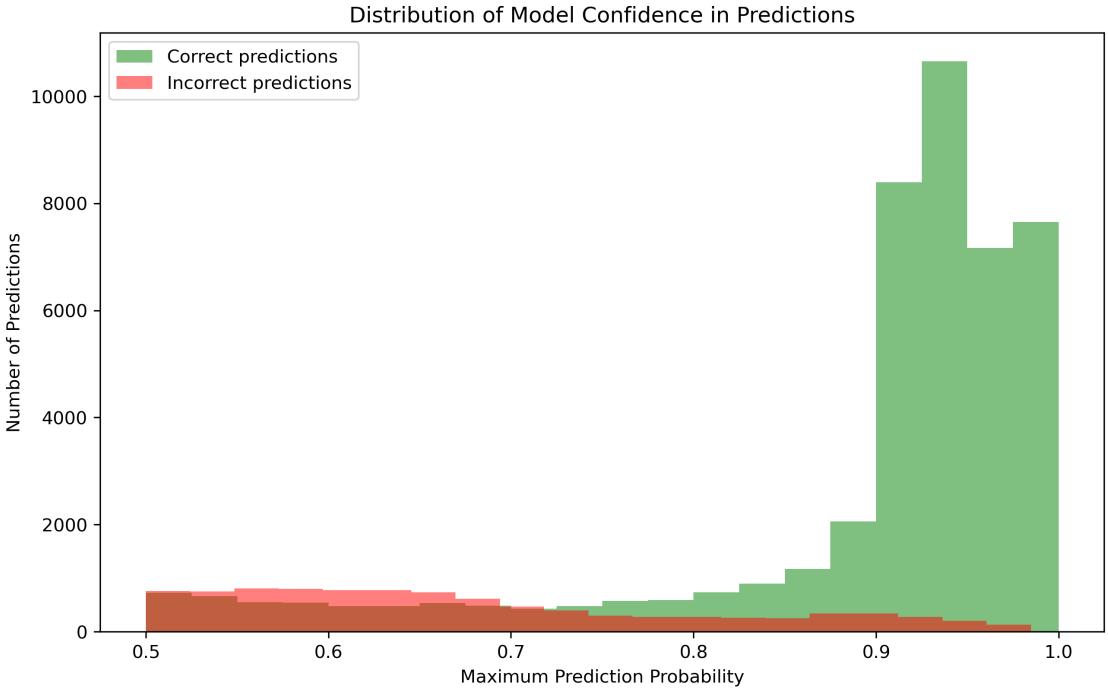
To validate the importance rankings using a model-agnostic approach, we calculated permutation importance scores. In Figure 2, we observe similar dominant features, including std_AF, mean_SD, and std_SD, reinforcing the consistency of our findings. The error bars show the variability in importance scores across permutations, giving us additional confidence in these rankings. Again, autoencoder features were absent from the most informative set, emphasizing that the model's predictive power relied more heavily on engineered spatial statistics rather than learned embeddings.

(3) Confusion Matrix for Test Set Performance



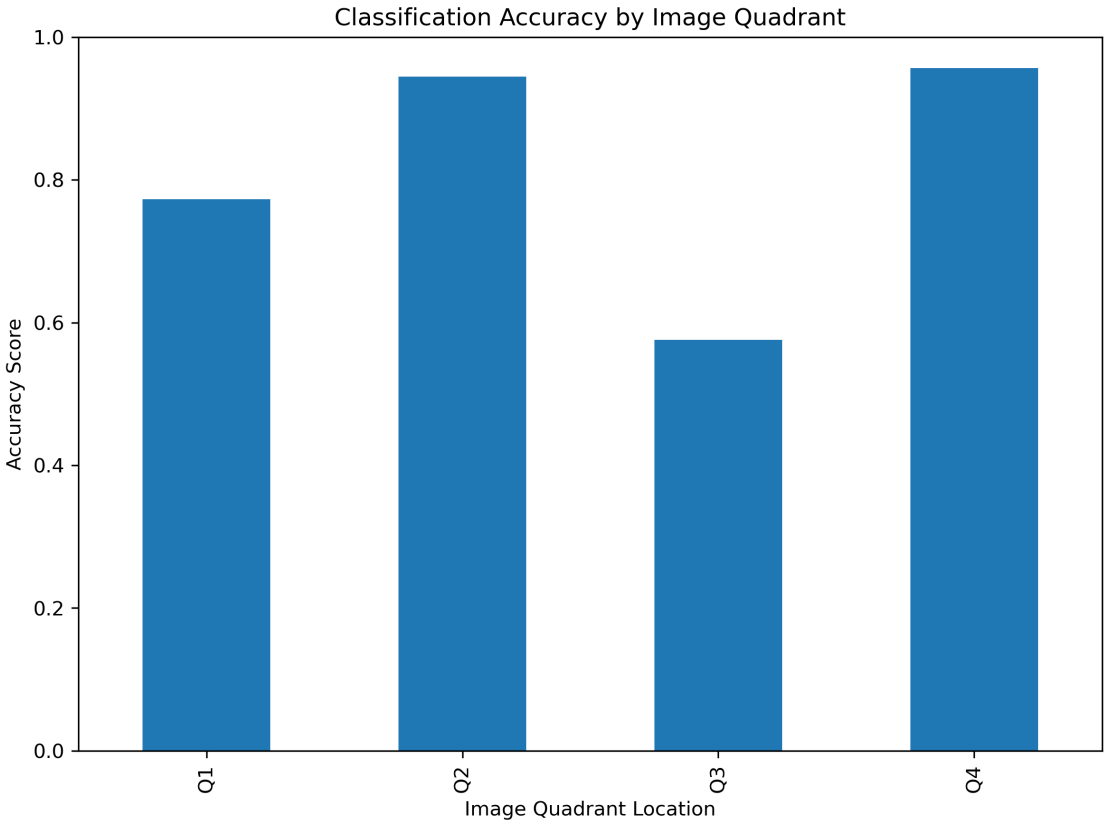
The confusion matrix (Figure 3) presents a clear breakdown of correct and incorrect predictions on the test set. The model accurately predicted 29,083 true negatives and 16,167 true positives, but also produced 4,445 false positives (non-cloud pixels incorrectly classified as clouds) and 5,077 false negatives (cloud pixels incorrectly classified as non-clouds). Notably, the false negative rate was slightly higher, indicating that the classifier sometimes struggled to detect thin or low-opacity clouds. This suggests that certain cloud types—possibly those with low reflectance or weak spectral contrast—were harder to classify correctly.

(4) Model Confidence and Prediction Probability Distribution



To understand the model's certainty in its predictions, we examined the distribution of prediction probabilities (Figure 4). The majority of correct predictions clustered around high probability values (>0.9), indicating strong classifier confidence in accurate classifications. In contrast, misclassified pixels were disproportionately found in the mid-confidence range (0.5–0.7), suggesting that borderline cases—such as mixed cloud/non-cloud pixels or faint cloud formations—were more challenging for the model. This pattern suggests that a confidence thresholding approach could help identify uncertain predictions, flagging them for manual review or further processing. Additionally, it demonstrates good model calibration, indicating that prediction confidence serves as a reliable proxy for uncertainty. Consequently, low-confidence predictions could be systematically flagged for human verification or additional computational refinement in an applied pipeline.

(5) Spatial Analysis: Accuracy by Quadrant



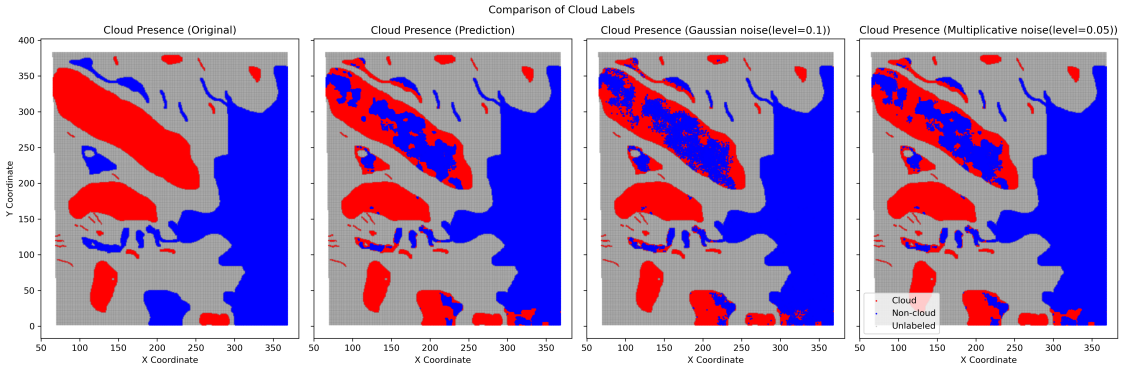
To further assess spatial variation in model performance, we analyzed accuracy stratified by image quadrant (Figure 5). The results revealed noticeable differences: Quadrants Q2 and Q4 achieved near-perfect accuracy (95%), whereas Q3 exhibited the lowest accuracy (65%). Upon further inspection, Quadrant Q3 contained more surface water and shadowed terrain, which might contribute to higher misclassification rates. The model's weaker performance in Q3 suggests that spectral and textural variations in certain regions affect classification accuracy. This emphasizes the importance of incorporating region-aware validation techniques and potentially augmenting training data with more diverse terrain types. Importantly, this analysis demonstrates the need for geographically-aware model validation strategies and potentially region-adaptive post-processing techniques to mitigate uneven performance across image space.

(6) ROC Curve Analysis

To further evaluate the classifier's ability to distinguish between classes, we plotted Receiver Operating Characteristic (ROC) curves (Figure 6). The curve for the cloud class shows strong performance, with an Area Under the Curve (AUC) of 0.94, indicating excellent separability between cloud and non-cloud pixels. The curve rises sharply toward the top-left corner, reflecting a high true positive rate with a low false positive rate across a range of thresholds. This reinforces earlier findings from the confusion matrix and confidence analysis, demonstrating that the model reliably identifies cloud regions while maintaining a low rate of false alarms. Overall, the ROC curve provides additional evidence of the Random Forest model's robust classification performance.

4.4 Generalization to Unlabeled Data and Stability Analysis

To evaluate how well our cloud classification model generalizes to future data without expert labels, we conducted a stability analysis by introducing controlled noise into key features and analyzing prediction consistency. Using the top 15 most important features identified through feature selection, we applied two types of perturbations: additive Gaussian noise (σ=0.1) to simulate sensor noise and multiplicative noise (σ=0.05) to model proportional measurement errors. The model was tested on three datasets—the original dataset, the Gaussian-noised dataset, and the multiplicative-noised dataset—to assess its sensitivity to small variations in feature values. The results showed minimal changes in cloud and non-cloud classifications across perturbed datasets, demonstrating strong prediction stability. Additionally, as a sanity check, we ran our classifier on unlabeled images to examine whether the predicted cloud regions appeared reasonable. The model produced spatially coherent cloud structures that aligned well with expected cloud patterns, reinforcing its ability to generalize beyond labeled training data. These findings suggest that the model is robust to small feature fluctuations and has the potential to perform well in real-world scenarios where expert labels are unavailable, provided that the input feature distribution remains consistent with the training data.



4.5 Conclusion and Results

To evaluate the generalization of our model, we tested all combinations of training on two labeled images and testing on the remaining third. This cross-image validation strategy ensures that each labeled image served as a test set exactly once, providing a fair assessment of performance across distinct spatial regions. The resulting classification accuracies were 0.81, 0.96, and 0.84, yielding a mean accuracy of 0.87. Notably, the three labeled images differ substantially in both cloud morphology and surface characteristics—ranging from dense, contiguous cloud masses to more dispersed or ambiguous formations. Training and testing on a single image could therefore introduce bias due to limited spatial and visual variability. To mitigate this, we applied the same model type—Random Forest—to all three configurations, while tuning hyperparameters separately for each training set. We report the average accuracy across these test cases to reflect this variability and to provide a more representative measure of the model family's overall performance. This approach offers a robust estimate of generalization capability, demonstrating that our model is adaptable to diverse Arctic cloud scenes rather than being overfitted to a specific spatial domain.

5. Bibliography

Shi, Tao, Bin Yu, Eugene E. Clothiaux, and Amy J. Braverman. 2008. Daytime Arctic Cloud Detection Based on Multi-Angle Satellite Data With Case Studies.

A. Academic Honesty

A.1 Statement

We confirm that this report represents the collaborative work of our entire group. All analysis methods and procedures were jointly designed and executed. The text, figures, and research process have been documented transparently to ensure reproducibility. Any references to others' work have been properly cited.

Research integrity is fundamental to academic progress. While scholarship builds on prior knowledge, every study must uphold truthfulness, reliability, and originality. Irreproducible methods or unattributed work undermine trust and devalue collective scholarly efforts. As a team, we affirm our commitment to transparency, respect for intellectual contributions, and accountability for maintaining ethical standards. Each member has ensured that our work is original, properly cited, and advances understanding of Arctic cloud detection through honest collaboration.

A.2 LLM Usage

ChatGPT was used as a coding assistant for syntax validation and visualization enhancements, specifically for refining graph color schemes. All analytical decisions, model implementations, and evaluations were performed independently by the authors.