Imbalanced Multi-layer Cloud Classification with Advanced Baseline Imager (ABI) and CloudSat/CALIPSO Data

Leah Ding, Roberto Corizzo
Department of Computer Science, American University
Washington, DC, USA
{ding, rcorizzo}@american.edu

Colin Bellinger
National Research Council of Canada
Ottawa, Canada
colin.bellinger@nrc-cnrc.gc.ca

Nancy Ching, Spencer Login, Rodrigo Yepez-Lopez
Department of Computer Science, American University
Washington, DC, USA
{nc0372a, sl4708a, ry4789a}@student.american.edu

Jie Gong, Dong L. Wu
NASA Goddard Space Flight Center
Greenbelt, MD, USA
{jie.gong, dong.l.wu}@nasa.gov

***Abstract*—Clouds at different altitudes play different roles in Earth's climate. Comprehensive understanding of overlapping clouds is important for climate and weather prediction. The East Pacific region is where El Niño and La Niña originate and where multi-layer clouds frequently occur. The overlap of clouds at different altitudes in this region increases the classification complexity for cloud-based climatological studies. Unlike prior work in cloud layer classification that assumes single layer or two-layer of clouds, in this work, we consider multi-layer cloud classification with 8 cloud-level classes (clear-sky, high, middle, low, high+middle, high+low, middle+low, high+middle+low). We develop and analyze machine learning models on features extracted from satellite images from the East Pacific regions collected by GOES Advanced Baseline Imager (ABI). These are used to classify CloudSat/CALIPSO observed multi-layer clouds. Due to the imbalanced nature of the data, we investigate the adoption of conventional resampling methods, as well as deep learning methods with data augmentation. In our experiments, we utilize the random forest classifier and Multilayer perceptron classifier with data augmentation methods to reduce the class imbalance during training. With these approaches, we achieve a classification accuracy of 83.6% without exploiting any ancillary information.**

***Index Terms*—Multi-layer cloud classification, class imbalance, machine learning, deep learning, data augmentation.**

I. Introduction

It is climatologically important to understand the vertical distribution of clouds in near-Equatorial regions, such as the Eastern Pacific. However, the current ability to measure the vertical structure of clouds is severely limited by the lack of spatial and temporal satellite image coverage. For example, the NASA Earth observation satellite CloudSat and the Cloud-Aerosol Lidar and Infrared Pathfinder Satellite Observations (CALIPSO) satellite provide the global vertical information on clouds at nadir. The temporal gap between revisit with these systems, however, is long. As a result, they can not be utilized for real-world weather studies.

On the other hand, the Advanced Baseline Imager (ABI) onboard the GOES satellites monitor the pan-disk clouds at very high temporal and spatial resolution. The availability of such passive image data provides an opportunity to study vertical structure of clouds. Classifying overlapping cloud layers based on passive satellite image data, however, is an extremely challenging task. This is because the satellite imagery cannot penetrate through thick and precipitating clouds. In addition, clouds at different altitudes may vertically overlap with each other, thus, low-level clouds may be hidden from the view of passive satellites.

Being able to discriminate between multi-layer cloud classes has many applications in weather forecasting and climate research. Since the available data is extremely imbalanced, however, the challenge of inducing accurate classification models from such data is significant. As a result, to the best of our knowledge, there are no existing studies that investigate learning multi-layer cloud classification models from available imbalanced multi-layer cloud data.

In this paper, we study the problem of multi-layer cloud classification using only passive satellite observations (i.e., ABI data). We employ different machine learning models with ABI data as features and utilize the information from CloudSat/CALIPSO as the ground truth targets. Our contribution is twofold. First, we tackle the challenging task of multi-layer cloud classification. We formalize this as an 8-classes classification problem (clear-sky, high, middle, low, high+middle, high+low, middle+low, high+middle+low) instead of adopting the conventional, and more limited approach, of 3-classes (low, mid, high). Second, we introduce use of deep learning and data augmentation as a state-of-the-art approach to handle imbalanced multi-layer cloud classification with the objective of improving classification accuracy. We expect our results to be a useful reference for future studies on cloud classification from passive ABI and active CloudSat/CALIPSO data.

The paper is structured as follows. Section II summarizes related works for cloud classification and imbalanced data augmentation. Section III describes the analyzed data and our methods. Section IV describes the experimental settings, and discusses the results obtained by our study. Finally, Section V concludes the paper outlining directions for future work.

II. Related Work

A. Cloud classification methods

Since clouds present relatively continuous properties due to similar atmospheric conditions, simple layer clouds (e.g., upper-layer ice cloud, lower-layer water cloud, or overlapping two-layer clouds) can be classified using statistical extrapolation methods based on image observations [1]. However, the problem of classifying overlapping clouds with granular elevation classes (e.g., high, middle, low) and multilayer (e.g., high+middle, high+low, middle+low, high+middle+low) have not been well studied for two major reasons. First, it is very challenging to accurately classify cloud elevation based on satellite images captured by the GOES Advanced Baseline Imager (ABI). This is due to the non-linear relationship between cloud elevation and cloud visual appearances in imagery data. Secondly, it is very challenging to learn from imbalanced multi-layer cloud data.

As information on overlapping cloud properties is essential for climate and weather prediction, increasing attention has been paid to the detection of overlapping clouds and the retrieval of their properties [1]–[5]. Recent studies have utilized physics-based numerical and statistical methods to classify clouds. [3] used a threshold based numerical detection method for two-layer clouds (overlapping two-layer versus single-layer cloud). [1] proposed a statistical extrapolation algorithm for retrieving the cloud top heights of two-layer clouds.

Such numerical and statistical methods, while much simpler to implement, show limited potential for classifying more complex multi-layer cloud data (e.g., 8 classes of multi-layer clouds). Effective machine learning-based approaches have been introduced to study cloud type categories(e.g., cirrus, cumulus, stratus, etc.) [6]–[10], detect the presence of snow/ice in clouds, classify thick versus thin clouds [11], predict cloud top height [4] and predict the macro-physical parameters of clouds [12].

Differently from prior work, this paper focuses on the 8-class multi-layer cloud classification problem. Moreover, we utilize state-of-the-art machine learning classifiers and data augmentation methods. In this work, the 8 classes are enumerated as (1) clear-sky, (2) low cloud, (3) middle cloud, (4) overlapping low and middle clouds, (5) high cloud, (6) overlapping low and high clouds, (7) overlapping middle and high clouds, and (8) overlapping low, middle and high clouds. Figure 1 illustrates the imbalances in the number of examples for each cloud layer in the utilized dataset. The figure shows that class 1 (clear sky) and class 2 (low cloud) dominate the other 6 classes (*i.e.,* classes 3 to 8 are minor classes.)

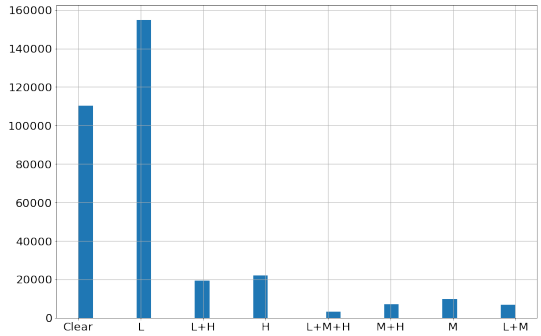

Fig. 1: Number of data samples in each class

B. Data Augmentation

Data augmentation methods for imbalanced data classification can be broadly categorized in two groups: model-agnostic methods, and methods specific to neural network models. Model-agnostic methods present the advantage of not depending on a specific classification model. Traditionally, the most wide-spread techniques to handle the class imbalance problem are resampling methods, such as random undersampling (RUS), random oversampling (ROS), and synthetic minority oversampling (SMOTE).

While RUS produces an balanced training set by removing training samples from the majority class, ROS produces a new training set by means of oversampling. In both cases, samples are randomly drawn with replacement from the original dataset. RUS and ROS are simple to implement and efficient to employ, however, they have significant limitations. ROS is known to cause the trained model to overfit the replicated samples, whilst the RUS can cause the learning model to underfit regions of the majority class where informative samples have been discard from [13]. Possible alternative undersampling approaches to mitigate this issue involve the adoption of support vectors [14].

The SMOTE technique was developed to reduce the need to discard potentially informative majority samples and avoid overfitting the minority class by utilizing interpolation rather than replication [15]. SMOTE generates new synthetic samples by means of random interpolation between instances in a dataset. Similar to RUS and ROS, SMOTE can be applied independently to each minority class in a multi-class setting.

SMOTE has produced positive results on a plethora of imbalanced classification problems, however, it also has well-documented limitations, such as suffering from high variance and generating noisy instances that encroach on majority class space [16]. These and other limitations have inspired numerous alternative synthetic oversampling algorithms [17].

The adaptive synthetic oversampling method (ADASYN) is designed address some of the weaknesses of SMOTE by perform minority class interpolation in a manner that accounts for the class density around the minority class instances [18]. In particular, given a minority class instance, the number of synthetic minority class instances generated via interpolation

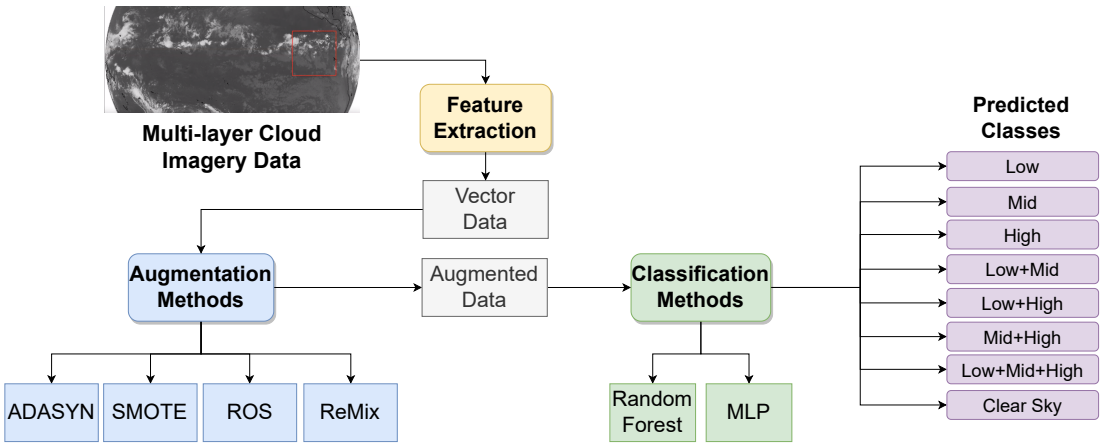

Fig. 2: Method workflow. Following imagery data acquisition and feature extraction, tabular (vector-based) data is fed to augmentation methods to address class imbalance. Subsequently, the classification task at hand is carried out with machine learning classifiers.

depends on the class distribution in the neighbourhood of the instance.

One possible limitation of model-free methods is that they focus on the static generation of augmented samples, which may represent a limitation in online and streaming data settings, where the entire dataset is not available at once.

In recent years, the supremacy of deep learning [19] has inspired novel augmentations techniques to improve generalization and ameliorate the problem training deep neural networks on imbalanced classification data [20]. Previous work, such as GAMO, Deep SMOTE, WGAN and C-VAE, utilized deep generative techniques to model and synthesize minority samples to balance the training set [21]–[24]. Alternatively, integrated data augmentation methods deep neural networks aim to efficiently improve predictive performacne and generalization. This form of adaption can function at the mini-batch level, thus, reducing the computations burdent and enabling the model to be trained incrementally as new data becomes available. Moreover, the authors in [25] proposed an imbalanced augmentation method that generates new samples in the feature and the target spaces, in order to improve the smoothness of the decision boundary, whilst improving the classification of the minority classes.

III. Data and Methods

A. Data

In this work, we focus on data collected in East Pacific (10°S-10°N, 90°W-110°W). This is shown in the red box in Figure 3. This area is often covered with multi-layer clouds (55% occurrence frequency), and it's very hard to separate the low-, middle- and high-level clouds from passive satellite image data (e.g., ABI). The ABI onboard the NOAA GEOS-R satellites monitor the pan-disk weather at very high temporal and spatial resolutions but cannot penetrate through thick and multi-layer clouds. We collect ABI observations from sensing-based images [26] that capture the Earth with 16 different spectral bands, including two visible channels, four

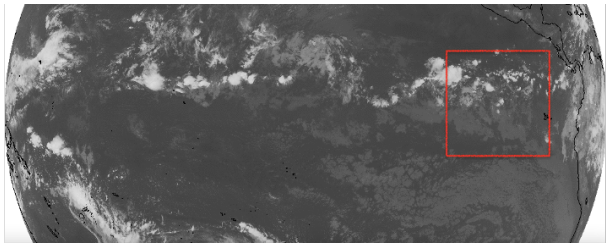

Fig. 3: Example of sensing-based image

near-infrared channels, and ten infrared channels. Different channels are different from one another in terms of resolutions (0.5km to 2km) and their sensitivities to different levels of clouds. We use all 16 channel features from ABI data in our study.

The synergistic combination of CloudSat and CALIPSO provides vertical profiles of clouds [27]. Figure 4 shows an example view of the vertical structure of multi-layer clouds from the earth surface with timestamp, latitude, and longitude. The multi-layer cloud classes are determined based on the height of clouds from the collocated CloudSat/CALIPSO data.

When CloudSat radar passes the region of interest, there is always a corresponding ABI pixel value as ABI data has very high temporal and spatial resolutions. We collocate CloudSat/CALIPSO observations with ABI pixel values with geographic location and timestamp from June 2019 to September 2020.

B. Data Augmentation Methods

In this subsection, we briefly summarize and formalize the data augmentation methods adopted in this study.

1) ROS: Given a training set D composed of two classes D_{maj} and D_{min}, where $|D_{\text{maj}}| \gg |D_{\text{min}}|$, ROS produces a new training set $D' = \{E \cup D_{\text{maj}}\}$, where E is a sample set randomly drawn with replacement from D_{min}. ROS can be

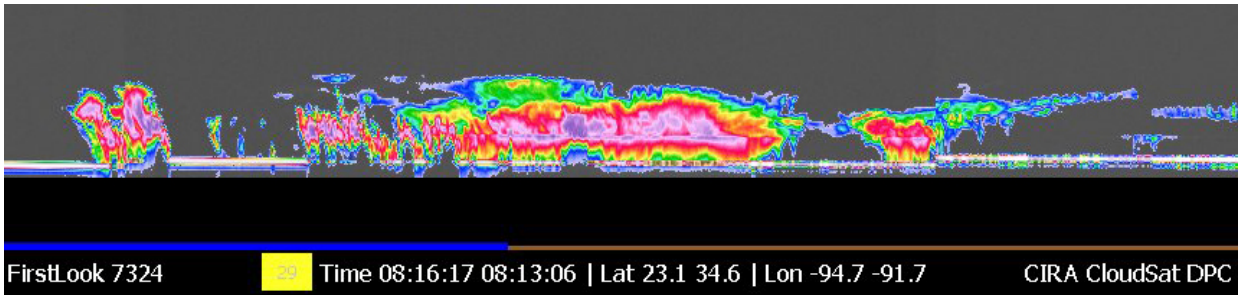

Fig. 4: Example of collocated CloudSat/CALIPSO cloud vertical observations

generalized to the multi-class setting by performing resampling on each class to achieve a balance across all classes in the training set.

2) SMOTE [15]: The SMOTE algorithm produces a balanced training set $D' = \{E \cup D_{\text{maj}}\}$, where E is composed of synthetic instances generated by random interpolation between instances in D_{min}. In particular, SMOTE randomly samples an instances $d_i \in D_{\text{min}}$, finds the set K_{d_i} of k-nearest neighbours to d_i in D_{min}, $K_{d_i} = \text{knn}(d_i, k)$, randomly selects a neighbour $d_j \in K_{d_i}$, and performs linear interpolation between d_i and d_j. The interpolated sample, e, is computed as:

$$e = d_i - (d_j - d_i) \times \delta, \tag{1}$$

where δ is uniformly and randomly sampled from the interval $(0, 1)$. This process is applied repeated to produce the set E, such that E has the desired level of balance with the majority class. The proceduce can be generalized to the multi-class setting by applying it independently to each minority class.

3) ADASYN [18]: The ADASYN algorithm produces a balanced training set $D' = \{E \cup D_{\text{maj}}\}$. To populate E with class density adapted synthetic minority samples, ADASYN requires the user to specify the total number of synthetic minority samples, G, to generate. Typically, $G = |D'_{\text{min}}|$. Based on G, it calculates the required number of instances, g_i, to be generated for the real minority instance $d_i \in D_{\text{min}}$. The value g_i is determined based on the class density around d_i as:

$$r_i = \frac{\Delta_i}{K}, \tag{2}$$

where K is the number of nearest neighbours to consider and Δ_i is the number of K-nearest neighbours of d_i in D that belong to the majority class. Therefore, r_i is the portion of majority class instances in the K-neighbourhood of d_i. The value r_i is normalized by dividing by the sum of all $r_j \in 1..|D_{\text{min}}|$ and denoted $\hat{r}_i$. Finally, the number of synthetic samples to generated for d_i is calculated as:

$$g_i = \hat{r}_i \times G. \tag{3}$$

For each minority sample $d_i \in D_{\text{min}}$, g_i synthetic minority samples are generated according to the minority class interpolation procedure presented in the SMOTE algorithm.

4) ReMix [25]: This method follows the same training strategy as MixUp [28]. The MixUp algorithm is a training strategy for deep learning that address generalization rather than class imbalance. This is achieved via data augmentation. The data augmentation is performed by forming unsupervised convex-combinations of mini-batch data and class labels. In particular, for each mini-batch of real training data B^i, a substitute augmented mini-batch, $B^{i\prime}$ is generated for training as follows:

$$B_x^{i\prime} = \lambda B_x^i + (1 - \lambda) B_x^i[\text{IDX}] \tag{4}$$

$$B_y^{i\prime} = \lambda B_y^i + (1 - \lambda) B_y^i[\text{IDX}], \tag{5}$$

where λ is randomly sampled from the beta distribution at the beginning of each episode, IDX is a random permutation of mini-batch order, and $B_x^{i\prime}$ and $B_y^{i\prime}$ are the augmented feature and one-hot encoded class labels. The label augmentation results in soft-labels, which have been demonstrated to improve calibration. The augmented data is utilize to train the deep model according to the principle of Vicinal Risk Minimization, which has been shown to improve generalization [29].

As demonstrated by the authors of ReMix, however, MixUp is negatively impacted by class imbalance. ReMix conducts a weighted sampling of each mini-batches to ensure the classes are approximately balanced, and generates augmented balanced mini-batches from unsupervised convex-combinations. This method enables the balancing in mini-batches to reduce predictive bias, whilst it improves calibration with soft-label training, and reduces the risk of overfitting the augmented minority class. This is because ReMix efficiently enables the method to generate a potentially infinite number of synthetic training samples. ReMix can be applied seamlessly to any number of classes.

C. Classification Algorithms

1) Random Forest (RF): This is an ensemble classification algorithm formed of many decision trees [30]. In order to increase the robustness of the ensemble, randomness is infused into the individual trees in two ways. Each decision tree in the random forest is build from a bootstrap sample (subset with replacement) of the training set. In addition, each split in the tree is generally based on a random subset of the complete feature set. By combining the predictions from several randomized decision trees, random forests can often

produce better generalization and predictive performance than the individual trees.

2) Multilayer Perceptron (MLP): This classification algorithm is formed of a fully connected artificial neural network [31]. The classifier takes a feature vector or tensor as the input. The input is mapped through multiple fully connected hidden layers, which contain hidden weights, to produce a classification at the output layer. In each hidden layer, a non-linear activation function, such as sigmoid or rectified linear unit (relu), is applied to facilitate a non-linear model. The output of the final hidden layer is combined and passed through the softmax function to produce the class prediction. The weights of the model are trained in a supervised manner to produced the desired classification via stochastic gradient descent and the backpropation algorithm. In this work, the cross-entropy loss is minimized during training.

IV. Experiments

Our preliminary experiments involved multiple classification algorithms, including multi-label SVMs, MLP, Decision Trees, and Random Forest. These experiments highlighted that, among the vanilla models, Random Forest and MLP achieved the highest overall accuracy. For this reason, we selected them to perform additional experiments in combination with resampling methods, including SMOTE, RandomOverSampler and ReMix. Experiments are performed using a fixed holdout validation procedure (80% training, 20% testing).

A. Hyperparameters

The HyperOpt optimizer [32] was used to fine-tune all models. Specifically, for Random Forest we first optimized parameter selection with the following space configuration: *n_estimators: [100, 200, 300, 400,500,600], min_samples_split: [2, 5, 10, 15, 100], min_samples_leaf: [1, 2, 5, 10] criterion: [gini, entropy],* [32]. The best parameters found were: *criterion = entropy, n_estimators = 800, min_samples_split = 3*. All *min_samples_split* options were found to be hindering the accuracy, therefore the default parameter (*min_samples_leaf = 1*) was used.

For MLP models, the following configuration was used: *activation=logistic, solver=adam, learning_rate=adaptive, alpha=1.97e-05, batch_size=666, learning_rate_init=0.001.* These results were found by specifying a range of values for the numerical hyperparameters and listing the choices for the categorical hyperparameters.

For the *activation* hyperparameter, the following options were considered: *identity, logistic, tanh, relu*. For the *solver* hyperparameter, the following options were considered: *lbfgs, sgd, adam*. For the *learning_rate* hyperparameter, the following options were considered: *constant, invscaling, adaptive*.

For all the numerical hyperparameters, a range was specified for each hyperparameter. The range for the *alpha* hyperparameter was from 0.00001 to 0.001. This was to explore the values around the default value, which was 0.0001. The range for the *batch_size* hyperparameter was from 500 to 1000. This was because the default value, 200, was too small relative to our very large sample size. The range for the *learning_rate_init* was from 0.001 to 0.01. The optimized value ended up rounding down to the default value, which is 0.001.

B. Metrics

In our study, we adopt standard metrics in machine learning-based classification, such as Precision (P), Recall (R), and F-Measure ($F1$), defined as:

$$P = \frac{T_p}{T_p + F_p}; \quad R = \frac{T_p}{T_p + F_n}; \quad F1 = 2 \times \frac{P \times R}{P + R},$$

where T_p is the number of true positive, and F_p is the number of false negatives. To properly consider class imbalance, we adopt the weighted variants of Precision, Recall, and F-Measure, where values are calculated for each label, and their average is weighted by support (the number of true instances for each label).

In addition, we adopt the Balanced Accuracy (BA) measure, which allows us to better assess the classification accuracy of models considering the imbalanced setting we are dealing with. It is computed as the average Recall obtained on each class:

$$BA = \frac{1}{C} \sum_{c=1}^{C} R_c,$$

where C is the total number of classes in the dataset.

C. Results and discussion

1) Random Forest: When observing the overall accuracy, the baseline Random Forest (no data augmentation) model set to a single train/test split achieved an overall high accuracy (82.4%) but when noting the balanced accuracy score, the model only produces a balanced score of 73.3%. Looking at just the baseline model, as shown in 5, the model was easily able to differentiate clear skies from class Low+Middle+High, overlapping cloud observations that were recorded at low, middle and high, while having relatively more difficulty predicting the minority classes. Across all resampled models, minority classes also saw significantly better accuracy once resampled. For example, in the baseline Random Forest model, prediction accuracy for the smallest class L+M+H was 58.3% without data augmentation, and it was increased to 64.1% with ROS, 65.7% with SMOTE, and 64.8% with ADASYN.

The accuracy was increased from 61.8% to 68.7% with ROS, to 71.4% with SMOTE, to 71.7% with ADASYN for overlapping low and middle clouds. The accuracy for majority class remained relatively the same with ROS. For example, the accuracy for Clear-sky class was 80.1% by baseline Random Forest model and 80.5% with ROS. The the accuracy for the low cloud class was 88.3% by baseline Random Forest model, 88.3% with ROS. The accuracy for majority class remained relatively the same with the exception of the low cloud class (the largest class) where the model with SMOTE and ADASYN predicted less accurately than the baseline model,

TABLE I: Experimental results: Random Forest with no data augmentation (Baseline), with ROS, SMOTE, and ADASYN.

Metric	Baseline	ROS	SMOTE	ADASYN
Accuracy	0.824	0.836	0.827	0.827
Balanced Accuracy	0.733	0.773	0.773	0.774
Precision	0.823	0.835	0.829	0.828
Recall	0.824	0.836	0.827	0.827
F1	0.822	0.835	0.828	0.827

TABLE II: Accuracy for each class: Random Forest with no data augmentation (Baseline), with ROS, SMOTE, and ADASYN.

Class (Sample Ratio)	Baseline	ROS	SMOTE	ADASYN
L+M+H (1%)	0.583	0.641	0.657	0.648
M+H (2%)	0.686	0.729	0.733	0.747
L+M (2%)	0.618	0.687	0.714	0.717
M (3%)	0.811	0.846	0.839	0.834
L+H (6%)	0.679	0.750	0.738	0.738
H (7%)	0.810	0.842	0.822	0.830
Clear-sky (33%)	0.801	0.805	0.820	0.834
L (46%)	0.883	0.883	0.857	0.844

likely due to more minority class samples muddying the class features. The variety of prediction choices also remained true. . For example, the accuracy for Clear-sky class was 80.1% by baseline Random Forest model, 82.0% with SMOTE, and 83.4% with ADASYN. The the accuracy for the low cloud class was 88.3% by baseline Random Forest model, 88.3% with ROS. There is a 2.6% decrease in low cloud accuracy with SMOTE and 3.9% decrease with ADASYN.

Using ROS, both accuracy and balanced accuracy improved from the baseline Random Forest model and is the highest score. Variety in class prediction also remains true while class prediction accuracy is also slightly improved. Utilizing the hyperparameter optimization from Hyperopt, combining the optimized Random Forest with the ROS resulted higher Recall, Precision and F1 metrics. The confusion matrices for Random Forest in combination with the different augmentation methods are reported in Figure 5.

2) Multilayer Perceptron (MLP): The overall highest accuracy score achieved by the MLP classifier was 72.9%. This score was achieved by the MLP model with ReMix. The overall accuracy of the baseline MLP model with no data augmentation was 72.6%. It must be noted that the balanced accuracy score for this model was the lowest, at 45.6%, when compared to the MLP models with resampling methods applied.

As shown in Table II, resampling methods significantly boosted accuracy for the smallest class L+M+H, i.e., accuracy was 4.6% without data augmentation, and it was increased to 65% with ReMix, 42.7% with ROS, 50.1% with SMOTE, and 39.1% with ADASYN. They show improvements over all other minority classes as well. For example, the accuracy for M+H class was 21.7% without data augmentation, and it was increased to 72% with ReMix, 52.9% with ROS, 51.5% with SMOTE, and 58.1% with ADASYN. The confusion matrices for MLP in combination with the different augmentation methods are reported in Figures 6 and 7.

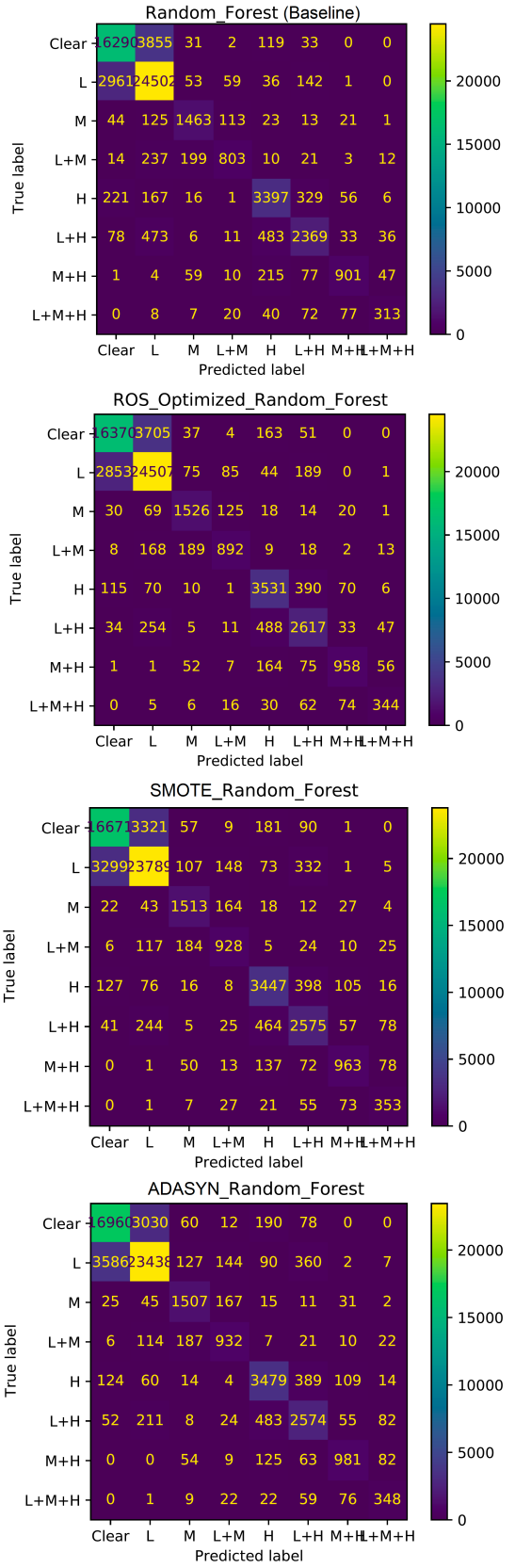

Fig. 5: Confusion matrices for the Random Forest classifier. From top to bottom, the matrices correspond to Random forest with no data augmentation (baseline), with ROS, SMOTE, and ADASYN.

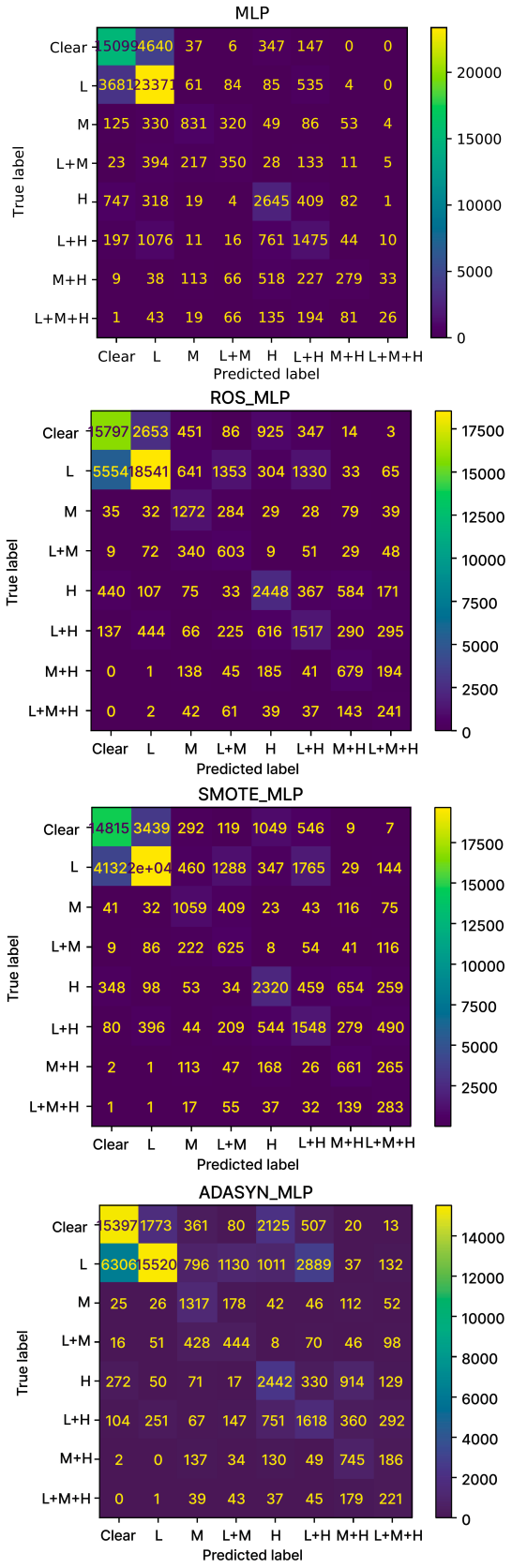

Fig. 6: Confusion matrices for the MLP classifier. From top to bottom, the matrices correspond to MLP with the following pre-processing: none, ROS, SMOTE, ADASYN.

TABLE III: Experimental results: MLP with no data augmentation (Baseline), with ReMix, ROS, SMOTE, and ADASYN.

Metric	Baseline	ReMix	ROS	SMOTE	ADASYN
Accuracy	0.726	0.729	0.677	0.675	0.621
Balanced Accuracy	0.456	0.701	0.579	0.570	0.554
Precision	0.715	0.755	0.689	0.718	0.703
Recall	0.726	0.729	0.717	0.675	0.621
F1	0.717	0.734	0.677	0.691	0.639

TABLE IV: Accuracy for each class: MLP with no data augmentation (Baseline), with ReMix, ROS, SMOTE, and ADASYN.

Class (Sample Ratio)	Baseline	ReMix	ROS	SMOTE	ADASYN
L+M+H (1%)	0.046	0.65	0.427	0.501	0.391
M+H (2%)	0.217	0.72	0.529	0.515	0.581
L+M (2%)	0.301	0.67	0.519	0.538	0.382
M (3%)	0.462	0.79	0.707	0.589	0.732
L+H (6%)	0.411	0.61	0.423	0.431	0.451
H (7%)	0.626	0.75	0.579	0.549	0.578
Clear-sky (33%)	0.745	0.82	0.779	0.731	0.759
L (46%)	0.840	0.69	0.666	0.707	0.558

V. Conclusion

In this paper, we explored the problem of multi-layer cloud classification using only passive satellite observations (ABI data). Different combinations of machine learning models and data augmentation methods were considered in order to effectively deal with the imbalanced multi-class classification task. Our most accurate classification results obtained combining random forest with data augmentation methods reveal an accuracy of 82.7%. Satisfactory but sub-optimal results were also extracted with multilayer perceptron and ReMix for data augmentation. This result was partially expected due to the tabular nature of the data, which makes it more difficult to extract an optimal neural network model represents, and an easier setting for random forest. Overall, we expect our results to be a useful reference for future studies on cloud classification from passive ABI and active CloudSat/CALIPSO data.

Our most promising future work involves applying deep

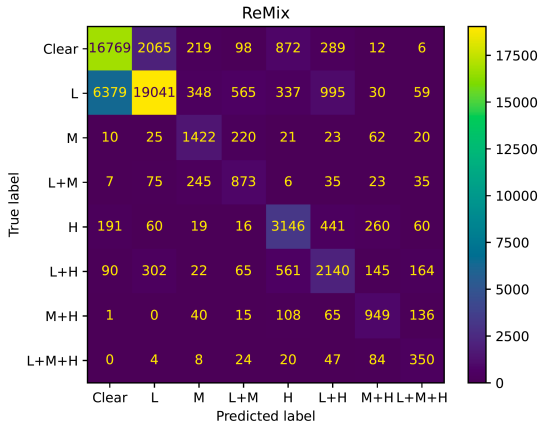

Fig. 7: Confusion Matrix of MLP classifier with ReMix data augmentation

learning methods to the raw satellite image data. In other image classification tasks, deep learning has been shown to significantly outperform models trained with human-engineered features, such as those used in this study. Thus, we plan to adapt deep learning-based augmentation methods to our multilayer cloud image datasets, as well as the investigation the use of semi-supervised learning approaches to leverage the plethora of partially labelled data available in this domain. Additional directions include addressing more complex machine learning tasks that consider the temporal and spatial dimension of the data, as well as multi-task approaches that combine classification and forecasting.

ACKNOWLEDGMENTS

This research is made possible by the generous support of National Science Foundation (Grant number 2150420), National Security Agency (Grant number H98230-22-1-0014), and American University as part of the SPIRAL/SPATIAL-Stats REU project.

REFERENCES

[1] Z. Tan, S. Ma, C. Liu, S. Teng, N. Xu, X. Hu, P. Zhang, and W. Yan, "Assessing overlapping cloud top heights: An extrapolation method and its performance," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–11, 2022.

[2] T. Wang, E. J. Fetzer, S. Wong, B. H. Kahn, and Q. Yue, "Validation of modis cloud mask and multilayer flag using cloudsat-calipso cloud profiles and a cross-reference of their cloud classifications," *Journal of Geophysical Research: Atmospheres*, vol. 121, no. 19, pp. 11–620, 2016.

[3] J. Wang, C. Liu, B. Yao, M. Min, H. Letu, Y. Yin, and Y. L. Yung, "A multilayer cloud detection algorithm for the suomi-npp visible infrared imager radiometer suite (viirs)," *Remote Sensing of Environment*, vol. 227, pp. 1–11, 2019.

[4] J.-F. Rysman, C. Claud, and S. Dafis, "A machine learning algorithm for retrieving cloud top height with passive microwave radiometry," *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, 2021.

[5] J. M. Haynes, Y.-J. Noh, S. D. Miller, K. D. Haynes, I. Ebert-Uphoff, and A. Heidinger, "Low cloud detection in multilayer scenes using satellite imagery with machine learning methods," *Journal of Atmospheric and Oceanic Technology*, vol. 39, no. 3, pp. 319–334, 2022.

[6] C. Zhang, X. Zhuge, and F. Yu, "Development of a high spatiotemporal resolution cloud-type classification approach using himawari-8 and cloudsat," *International Journal of Remote Sensing*, vol. 40, no. 16, pp. 6464–6481, 2019.

[7] J. Huertas-Tato, F. J. Rodríguez-Benítez, C. Arbizu-Barrena, R. Aler-Mur, I. Galvan-Leon, and D. Pozo-Vázquez, "Automatic cloud-type classification based on the combined use of a sky camera and a ceilometer," *Journal of Geophysical Research: Atmospheres*, vol. 122, no. 20, pp. 11,045–11,061, 2017. [Online]. Available: https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1002/2017JD027131

[8] J. Gan, W. Lu, Q. Li, Z. Zhang, J. Yang, Y. Ma, and W. Yao, "Cloud type classification of total-sky images using duplex norm-bounded sparse coding," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 10, no. 7, pp. 3360–3372, 2017.

[9] Z. Yu, S. Ma, D. Han, G. Li, D. Gao, and W. Yan, "A cloud classification method based on random forest for fy-4a," *International Journal of Remote Sensing*, vol. 42, pp. 3353 – 3379, 2021.

[10] L. Yu, X. Jun, S. Chun-Xiang, and H. Yang, "An improved cloud classification algorithm for china's fy-2c multi-channel images using artificial neural network," *Sensors*, vol. 9, 01 2009.

[11] N. Ghasemian and M. Akhoondzadeh, "Introducing two random forest based methods for cloud detection in remote sensing images," *Advances in Space Research*, vol. 62, 05 2018.

[12] Y. Yang, W. Sun, Y. Chi, X. Yan, H. Fan, X. Yang, Z. Ma, Q. Wang, and C. Zhao, "Machine learning-based retrieval of day and night cloud macrophysical parameters over east asia using himawari-8 data," *Remote Sensing of Environment*, vol. 273, p. 112971, 2022.

[13] P. Branco, L. Torgo, and R. P. Ribeiro, "A survey of predictive modeling on imbalanced domains," *ACM Computing Surveys (CSUR)*, vol. 49, no. 2, pp. 1–50, 2016.

[14] B. Krawczyk, C. Bellinger, R. Corizzo, and N. Japkowicz, "Undersampling with support vectors for multi-class imbalanced data classification," in *2021 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2021, pp. 1–7.

[15] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.

[16] C. Bellinger, C. Drummond, and N. Japkowicz, "Manifold-based synthetic oversampling with manifold conformance estimation," *Machine Learning*, vol. 107, no. 3, pp. 605–637, 2018.

[17] A. Fernández, S. Garcia, F. Herrera, and N. V. Chawla, "Smote for learning from imbalanced data: progress and challenges, marking the 15-year anniversary," *Journal of artificial intelligence research*, vol. 61, pp. 863–905, 2018.

[18] H. He, Y. Bai, E. A. Garcia, and S. Li, "Adasyn: Adaptive synthetic sampling approach for imbalanced learning," in *2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence)*. IEEE, 2008, pp. 1322–1328.

[19] S. Ryan, R. Corizzo, I. Kiringa, and N. Japkowicz, "Deep learning versus conventional learning in data streams with concept drifts," in *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*. IEEE, 2019, pp. 1306–1313.

[20] K. Ghosh, C. Bellinger, R. Corizzo, B. Krawczyk, and N. Japkowicz, "On the combined effect of class imbalance and concept complexity in deep learning," in *2021 IEEE International Conference on Big Data (Big Data)*. IEEE, 2021, pp. 4859–4868.

[21] S. S. Mullick, S. Datta, and S. Das, "Generative adversarial minority oversampling," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 1695–1704.

[22] D. Dablain, B. Krawczyk, and N. V. Chawla, "Deepsmote: Fusing deep learning and smote for imbalanced data," *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[23] W. Dai, K. Ng, K. Severson, W. Huang, F. Anderson, and C. Stultz, "Generative oversampling with a contrastive variational autoencoder," in *2019 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2019, pp. 101–109.

[24] Q. Wang, X. Zhou, C. Wang, Z. Liu, J. Huang, Y. Zhou, C. Li, H. Zhuang, and J.-Z. Cheng, "Wgan-based synthetic minority oversampling technique: Improving semantic fine-grained classification for lung nodules in ct images," *IEEE Access*, vol. 7, pp. 18 450–18 463, 2019.

[25] C. Bellinger, R. Corizzo, and N. Japkowicz, "Calibrated resampling for imbalanced and long-tails in deep learning," in *International Conference on Discovery Science*. Springer, 2021, pp. 242–252.

[26] "Geostationary satellite imargery dataset," https://www.ssec.wisc.edu/data/geo/#/animation, accessed: 2022-08-26.

[27] "Cloudsat/calipso dataset," https://www.cloudsat.cira.colostate.edu/data-products/2b-cldclass-lidar, accessed: 2022-08-26.

[28] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," *arXiv preprint arXiv:1710.09412*, 2017.

[29] O. Chapelle, J. Weston, L. Bottou, and V. Vapnik, "Vicinal risk minimization," in *Advances in neural information processing systems*, 2001, pp. 416–422.

[30] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[31] P. E. Hart, D. G. Stork, and R. O. Duda, *Pattern classification*. Wiley Hoboken, 2000.

[32] J. Bergstra, D. Yamins, and D. Cox, "Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures," vol. 28, no. 1, pp. 115–123, 17–19 Jun 2013.