NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

APPLICATIONS OF BAYESIAN NEURAL NETWORKS TO GLOBAL PRECIPITATION MEASUREMENT MISSION DATA

by

Sean C. Heslin

March 2021

Thesis Advisor: Scott Powell
Co-Advisor: Marko Orescanin

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			*Form Approved OMB No. 0704-0188*

1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** March 2021	**3. REPORT TYPE AND DATES COVERED** Master's thesis

4. TITLE AND SUBTITLE
APPLICATIONS OF BAYESIAN NEURAL NETWORKS TO GLOBAL PRECIPITATION MEASUREMENT MISSION DATA

5. FUNDING NUMBERS

6. AUTHOR(S) Sean C. Heslin

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey, CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
N/A

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release. Distribution is unlimited.

12b. DISTRIBUTION CODE
A

13. ABSTRACT (maximum 200 words)

Meteorological remote sensing efforts have advanced operational decision making and scientific research over the last half-century by providing high-quality global observations of the land, atmosphere, and ocean. The continued development of convolutional neural networks (CNNs) and Bayesian neural networks shows potential for allowing some of these datasets to be synthetically produced where they cannot be directly observed. In this thesis, global precipitation measurement mission (GPM) data is used to train a rain-type classification Bayesian CNN (BCNN) using passive microwave data. Additionally, regression CNNs and BCNNs are trained to predict precipitation using GOES-16 multispectral infrared data over a tropical maritime region. The rain-type classification BCNN shows a 17% improvement in accuracy over existing literature, and the regression models demonstrate a proof of concept in using GPM radar data and geostationary radiances to train skillful CNNs and BCNNs to predict radar reflectivity and rain rate. The experiments demonstrate both the promise of using these data sources to train accurate models and the possible advantages of using BCNNs to quantify and better understand prediction uncertainty for these applications.

14. SUBJECT TERMS
artificial intelligence, remote sensing, tropical meteorology, convolutional neural networks, CNN, passive microwave, geostationary, synthetic radar, global precipitation measurement mission, GPM, Bayesian convolutional neural networks, BCNN, rain-type classification, GOES-16

15. NUMBER OF PAGES
65

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	**18. SECURITY CLASSIFICATION OF THIS PAGE**	**19. SECURITY CLASSIFICATION OF ABSTRACT**	**20. LIMITATION OF ABSTRACT**
Unclassified	Unclassified	Unclassified	UU

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release. Distribution is unlimited.

APPLICATIONS OF BAYESIAN NEURAL NETWORKS TO GLOBAL PRECIPITATION MEASUREMENT MISSION DATA

Sean C. Heslin
First Lieutenant, United States Air Force
BS, University of Utah, 2017

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN METEOROLOGY

from the

NAVAL POSTGRADUATE SCHOOL
March 2021

Approved by: Scott Powell
Advisor

Marko Orescanin
Co-Advisor

Wendell A. Nuss
Chair, Department of Meteorology

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Meteorological remote sensing efforts have advanced operational decision making and scientific research over the last half-century by providing high-quality global observations of the land, atmosphere, and ocean. The continued development of convolutional neural networks (CNNs) and Bayesian neural networks shows potential for allowing some of these datasets to be synthetically produced where they cannot be directly observed. In this thesis, global precipitation measurement mission (GPM) data is used to train a rain-type classification Bayesian CNN (BCNN) using passive microwave data. Additionally, regression CNNs and BCNNs are trained to predict precipitation using GOES-16 multispectral infrared data over a tropical maritime region. The rain-type classification BCNN shows a 17% improvement in accuracy over existing literature, and the regression models demonstrate a proof of concept in using GPM radar data and geostationary radiances to train skillful CNNs and BCNNs to predict radar reflectivity and rain rate. The experiments demonstrate both the promise of using these data sources to train accurate models and the possible advantages of using BCNNs to quantify and better understand prediction uncertainty for these applications.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

ABI	Advanced Baseline Imager
BCNN	Bayesian convolutional neural network
BNN	Bayesian neural network
CNN	convolutional neural network
CO	Core Observatory
dBZ	decibels of radar reflectivity factor
DPR	Dual-frequency Precipitation Radar
FD	full disk
GMI	GPM microwave imager
GOES	Geostationary Operational Environmental Satellite
GPM	Global Precipitation Measurement
GPROF	Goddard profiling
GPU	graphical processing unit
IR	infrared
KL	Kullback-Leibler
LEO	low earth orbit
MAE	mean absolute error
MBE	mean bias error
MSE	mean squared error
NASA	National Aeronautics and Space Administration
NN	neural network
PDF	probability distribution function
PMW	passive microwave
ResNet	residual networks
TRMM	Tropical Rainfall Measurement Mission

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

I would like to thank my advisor, Scott Powell, for his tireless input and commitment to upholding high standards for my work. Additionally, my co-advisor, Marko Orescanin, was invaluable with his expert knowledge in computer science and advice. Without their patience, guidance, and encouragement, I would not be where I am today. I would also like to thank Benjamin Marsh for all of his hard work as we learned about neural networks and for his assistance with Bayesian models later on in my research. Finally, I would like to thank my amazing wife, Chloe, for her constant love and encouragement, without which I doubt I would have ever made it through the last year.

THIS PAGE INTENTIONALLY LEFT BLANK

DISCLAIMER

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the U.S. Government.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

The continued development of space-borne remote sensing in the meteorological field has been an invaluable resource to numerous research and operational applications. In the military, this data is not only utilized for situational awareness during operations, but it also is an integral part of the data assimilated into the modern weather models used for planning and humanitarian efforts. The high spatial and temporal coverage of these technologies provides valuable insight into the current conditions of the atmosphere at a much larger scale than in situ observations or fixed, ground-based remote sensors. Development in space-borne precipitation radar and passive microwave (PMW) sensors such as onboard the Tropical Rainfall Measurement Mission (TRMM; Kummerow et al. 2000) satellite, which was retired in 2013, and now the Global Precipitation Measurement (GPM; Hou et al. 2014) mission Core Observatory (CO) have significantly increased skill in rain-type classification and precipitation estimates globally. While they provide a much larger coverage area than ground-based sensors, space-borne instruments on low earth orbit (LEO) satellites provide infrequent data to any single area, making them more difficult to use operationally. Despite this limitation, further application of the data captured by these sensors can allow for extended utilization.

Precipitation is among the most difficult to observe quantities associated with the atmosphere. Even direct measurements of precipitation are limited by properties of the rain gauge or disdrometer used to make the observation. This is unfortunate because performance of numerical models of the atmosphere is often determined by the modeled spatial pattern and intensity of precipitation. Many important quantities are directly impacted in the presence of rainfall including, latent heating, ocean salinity, surface temperature, soil moisture, and humidity (Clayson 2019). Latent heat release by phase changes in precipitating clouds also impacts the global circulation and observed estimates of latent heating are inferred from estimates of precipitation (Schumacher et al. 2004). Precipitation estimates derived from active (e.g., radar) and passive (e.g., infrared and microwave satellite radiances) remote sensors are subject to large errors because the observations they obtain do not fully, if at all, account for the specific hydrometeor size

and/or shape distribution. While active remote sensors are at least sensitive to the size of the largest hydrometeors present in a volume, passive remote sensors must use complicated retrievals to convert a satellite radiance, which contains no information about the size of any hydrometeors, into a precipitation estimate (e.g., Randel et al. 2020).

Onboard TRMM was a downward pointing K_u-band precipitation radar (Kummerow et al. 2000), and GPM currently carries a dual-frequency K_u/K_a-band radar (Hou et al. 2014). Combined, the two wavelengths permit estimation of both liquid and frozen precipitation rates and hydrometeor sizes. Both satellites also house passive microwave detection instruments that observe radiation emitted by the atmosphere and the surface. The passive microwave observations provide rudimentary information about the vertical structure of the atmosphere, which can be leveraged to aid in precipitation estimation (Huffman et al. 2007). However, GPM only scans a limited footprint at any given time, and the return period at a single point on Earth is several days. From LEO, simultaneously collecting passive and active microwave observations is infeasible. On the other hand, geostationary satellite radiances from platforms such as the Geostationary Operational Environmental Satellite (GOES) or Himawari-8 provide observations of a large portion of Earth every 15 minutes. However, passive and active microwave observations cannot currently be effectively collected from geostationary orbit, and the visible and infrared (IR) radiances detected by sensors on GOES-16, GOES-17, or Himawari-8 individually contain no information about the vertical structure of clouds and the atmosphere or the size of precipitation hydrometeors. Currently, geostationary IR data is used to advect PMW derived rainfall rates forward and backward in time between passes as shown in (Joyce and Xie 2011), in order to gather a more cohesive picture of global rainfall. Despite this limited example, estimating precipitation using these instruments alone remains difficult.

In recent years, interest in artificial neural networks (NNs) in the meteorological community has accelerated as the technology has been applied to a wide range of applications. One form of NN, the convolutional neural network (CNN) has shown significant success with image recognition (e.g., Krizhevsky et al. 2012). Once effectively trained, a CNN can identify the small-scale components of an image, and use the

arrangement of those components to identify more complex features, ultimately allowing the model to accurately identify very complex structures contained in an image. To create a model that can successfully predict the correct result, a NN is trained from a large, labeled dataset that contains the input vectors and associated labels the model should predict. Through successive training on a dataset, the NN learns the patterns that correspond to a given output. Furthermore, new advancements in this field are constantly improving the skill and usability of NNs. One technique that is currently emerging is the use of Bayesian NNs (BNNs; Kendall and Gal 2017), which have the capability of improving accuracy while also showing uncertainties in predictions. When applied with a large enough dataset and a sufficiently complex model, NNs have the capability to achieve impressive results.

The meteorological community has increasingly adopted NNs for remote sensing applications. For example, Hilburn et al. (2021) recently trained a CNN on GOES radiances towards the prediction of ground-based radar composite reflectivity. This work was presumably able to demonstrate success at estimating radar reflectivity from multispectral geostationary radiances because the combination of radiances contained underlying features that highlighted information about the physical composition of clouds that is not obvious by viewing a single channel alone but could be learned by a neural network. Likewise, we presume that geostationary IR or LEO PMW radiances may be used as feature vectors to estimate space-borne radar observations or rain-type classification, respectively, by learning these underlying features.

In this thesis, utilization of NN techniques to enhance the remote sensing capabilities in the meteorological community will be investigated. A background of the remote sensing capabilities and their use cases is covered along with a more detailed explanation of the design of NNs. First, a simpler classification problem is attempted to enhance rain type classification from PMW data along with the demonstration of the emerging technique, Bayesian NNs. Finally, a regression model is trained to simulate space-borne radar data from geostationary satellite data. The successful enhancement of rain type classification and application of geostationary satellite data towards simulating space-borne radar signatures would be significant in operational forecasting and research.

Furthermore, it would demonstrate the feasibility of these techniques and allow for others to develop even more successful models in the future.

II. BACKGROUND

A. GLOBAL PRECIPITATION MEASUREMENT MISSION

The GPM mission is the successor to the TRMM, which was conducted from 1997 to 2015, and has continued and expanded on its mission to assess global precipitation. GPM is an international project with nearly a dozen LEO satellites in the constellation and is shown in Figure 1. These satellites contain varying sets of sensors to aid in mapping global precipitation, however all satellites have PMW sensors onboard. Launched in 2014, the GPM CO is the principal component of the GPM mission and gathers data in a region from 65°N to 65°S at all longitudes. The CO carries the advanced Dual-frequency Precipitation Radar (DPR) which can capture the three-dimensional structure of precipitation from orbit in both 13.6 GHz (K_u-Band) and 35.5 GHz (K_a-Band) bands at a grid spacing of 5km horizontally and 250m vertically within a 245km swath width (NASA 2020). The K_u -Band radar has an effective sensitivity down to 17 decibels of radar reflectivity factor (dBZ), while the K_a-Band radar is sensitive down to 13 dBZ (Iguchi et al. 2017). Therefore, each radar is capable of observing all but the lightest precipitation such as drizzle or rainfall in decaying convection. Additionally, the CO contains the GPM Microwave Imager (GMI), which passively observes 13 frequencies between 10 GHz and 183 GHz at varying resolutions in a swath that is 885km in width. The highly detailed precipitation information captured by the DPR as well as the GMI sensors allows for accurate rainfall estimates, and calibration of the rest of the constellation's PMW sensors (Huffman et al. 2007) to improve their estimates as well. This enables the GPM primary mission of continuous, global, precipitation estimates to be achieved.

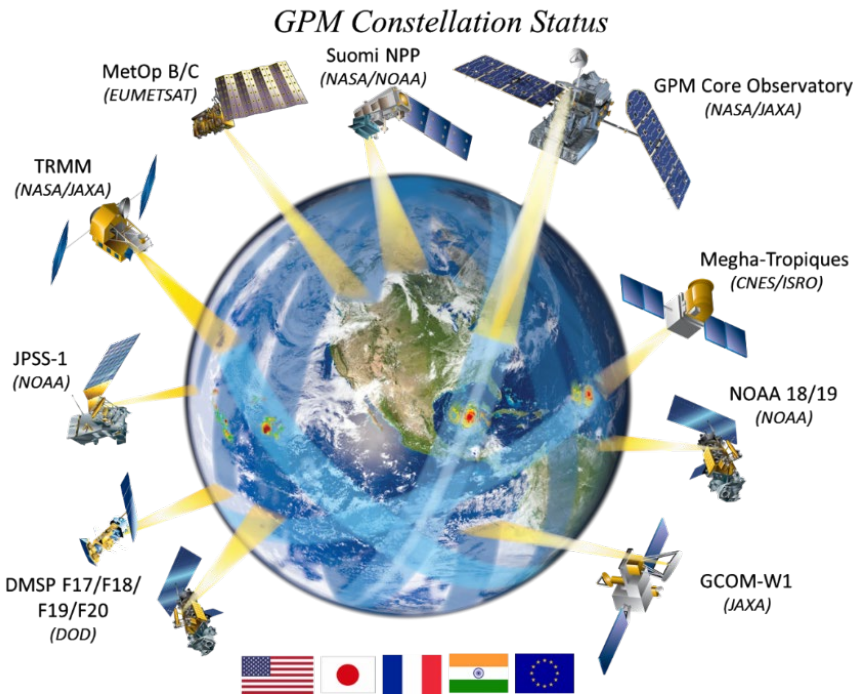


Figure 1. GPM Constellation as of April 2019. Source: NASA (2020).

B. RAIN TYPE CLASSIFICATION

Tropical rainfall can generally be classified into two regimes: convective and stratiform (Houze 1993). Stratiform rain generally produces smaller, but more numerous raindrops, while convective rainfall contains larger, less numerous drops (Thompson et al. 2018). Additionally, different latent heating profiles are produced by the two regimes, which can alter large scale circulations in the tropics (Schumacher et al. 2004; Takayabu and Tao 2020). Therefore, when estimating rainfall accumulation from remote sensing instruments, knowing the distribution of convective vs. stratiform precipitation is beneficial.

Since the significance of this classification was realized, several studies have endeavored to classify rainfall type and estimate precipitation using various space-borne and ground-based remote sensing datasets. In Steiner and Houze (1997), algorithms were devised to determine rain-type, storm structure, and precipitation amounts from ground-based radar data and was used to validate early TRMM data. Over time, these algorithms were enhanced to improve classification accuracy, especially for uncertain and shallow

convective classifications (Powell et al. 2016). Machine learning techniques have also been employed for classification on ground-based radar in Anagnostou (2004) as well as alongside more algorithmic approaches (Hagos et al. 2020; Wang et al. 2021). The DPR onboard the CO utilizes the high-resolution data captured by the K_a- and K_u-band radars combined with an advanced algorithm to produce a highly accurate depiction of rainfall classification (Awaka et al. 2016). However, the other satellites, shown in Figure 1, do not house a radar; therefore, classification and rain estimation must solely rely on their PMW observations. Determining the rainfall type of the precipitation captured by the PMW sensors is required to accurately predict the rainfall rate. The operational algorithm for determining the rainfall type for PMW observations in the GPM constellation is the Goddard profiling (GPROF) algorithm shown in Kummerow et al. (2015). Recently, approaches using NNs have shown increased accuracy compared to the GPROF algorithm (Petković et al. 2019). In Chapter III, a model is developed to build on this previous work.

C. GLOBAL PRECIPITATION ESTIMATION

Although numerous space-borne PMW sensors and ground-based radars are operating at any given time, the vast majority of the earth's surface is not directly observed by instruments capable of making accurate estimates of precipitation. The only data sources that provide consistent, large-scale observation of Earth are from geostationary satellites; however, since they only passively observe visible and IR radiation, it is difficult to obtain detailed information about the horizontal and vertical structure of clouds that would be useful for estimating precipitation. Creation of a precipitation radar at geostationary orbit has been theorized (Okazaki 2019), although an active radar sensor from that orbit is challenging and cost-prohibitive to engineer due to the extremely large antenna required to obtain useful spatial resolution. The concept of obtaining precipitation data from geostationary satellite data is not novel though. Arkin and Ardanuy (1989) and Huffman et al. (2001) demonstrated how to supplement LEO satellite data with geostationary IR data to generate precipitation maps at a spatial and temporal resolution required for climate monitoring. More recently, Behrangi et al. (2009) utilized PMW precipitation estimates to train a simple NN using multispectral geostationary data as input. Veillette et al. (2018) created a multi-source convolutional NN to simulate ground-based radar offshore of the

U.S. East Coast. This model utilized geostationary IR and visible data along with lightning data, numerical model output, and was trained with ground-based radar data as training labels. Impressive results were achieved with this highly complex model as they aimed to aid air-traffic controllers in data-sparse locations. As geostationary satellites, NNs, and computational power advance, the prospect for more accurate models does as well.

D. NEURAL NETWORKS

In recent years, NNs have proliferated throughout science and technology applications to the point where it is now common to hear the terms artificial intelligence, machine learning, and neural networks outside of these specific fields. While similar, these terms are not synonymous. Artificial Intelligence is simply any non-biological algorithm that demonstrates intelligence by following predefined rules. machine learning is any algorithm that can define those rules on its own given data. Finally, neural networks are a type of sophisticated machine learning algorithm that are built by combining layers of what are known as "neurons," which are a data structure partially inspired by biological neurons found in organisms.

1. Development and Structure

While the idea of NNs has been around for decades, it was not until advancement in computational power through graphical processing units (GPUs; Oh and Jung 2004) and continued development of improved model architectures (Glorot et al. 2011; He et al. 2015) that NNs came into the widespread use seen today. Furthermore, the creation of software libraries dedicated to developing machine learning models, such as TensorFlow and PyTorch, have simplified the process and availed this technology for use in a wide range of applications. TensorFlow is utilized for the creation of the models in this work.

Artificial neural networks are built off a simple neuron-like structure to generate highly complex models when many layers of neurons are connected. A single neuron takes multiple inputs, multiplies those inputs by corresponding weights, adds them together along with a bias value, and then inputs that value into an activation function to get the final output of the neuron. This output can then be the input for multiple other neurons

further along in the model. When training a model, the weights and bias of each neuron is adjusted.

To begin training a model, a loss function and a NN architecture must be selected, which both depend on the task the user is trying to accomplish. The loss function determines the training goal by defining a way to calculate the "cost" of an incorrect prediction. For example, when predicting continuous values, the most common loss functions are Mean Squared Error (MSE) and Mean Absolute Error (MAE). Both compare the model's predicted value and the dataset's actual label, with more incorrect predictions yielding a higher loss. The objective in training is to minimize the sum of the loss for all training values. The NN architecture is the specific arrangement of neurons in each layer one believes can accurately capture the relationship between the input and output data once the model is trained. Therefore, when determining the architecture to use, one must consider the number, size, and types of layers to incorporate in the model. NN models can vary greatly, from only a few layers of neurons to thousands of neurons across hundreds of layers.

During training, the weights of every neuron are adjusted through a process known as backpropagation. Since the difference between the predicted and actual values is known, the weights and biases of the neurons contained in the model can be adjusted back through the layers of the model to minimize the error. Therefore, as the model trains on more examples, the weights and biases of the neurons converge towards values that minimize the cost function, leading the model to become more accurate. Although, allowing this process to continue too long can lead to overfitting, where the model performs extremely well on the training data, but does not generalize well on external data. Due to this, a validation data set is often used during training which the model does not train on but is used to determine how well the model generalizes on additional examples.

Finally, there are several parameters that can be tuned in order to most effectively train a neural network, also known as hyperparameters. One of the most significant hypermeters when training a NN is learning rate. The learning rate determines how significantly the model changes the weights of the neurons when performing backpropagation. If the learning rate is too high, the weights may be adjusted too much,

causing the model to be unable to narrow in on the correct weights. Likewise, a learning rate that is too small may lead to the model learning slowly or not at all. The correct learning rate usually takes experimentation to discover and must be adjusted throughout training. Another significant hyperparameter is batch size. The batch size determines the number of samples to use each time the model performs backpropagation. Ideally, the model would use all samples in the dataset to perform backpropagation (e.g., via gradient decent method), however this is computationally intractable, and the backpropagation is calculated on subsets of data (e.g., stochastic gradient decent). The size of this subset is determined by the batch size, and is often between 200–500 samples, however different datasets and models can perform better with different sizes.

2. Convolutional Neural Networks

CNNs are a type of neural network architecture structured in a way that makes them effective at tasks involving images as the input. In a traditional "fully connected" NN, all the outputs of one layer are connected to each input of the next layer. This is an issue when using large images as an input, as this would be very computationally expensive. However, in a CNN, a kernel is used to break apart the image into what are known as a feature maps (Goodfellow et al. 2016). A kernel is a matrix, often 3x3 or 5x5, that looks at each portion of the image calculating a value for that subset. A kernel is another way to describe that, in the case of a 3x3 matrix, each artificial neuron for each layer only has 9 inputs, which are all located in a square on the input data. Multiple kernels with different weights can look at the image to create different feature maps. This process is visualized in Figure 2. Once the above process is completed for every pixel in the image, another convolutional layer can be applied to the feature map to identify more complex patterns in the image.

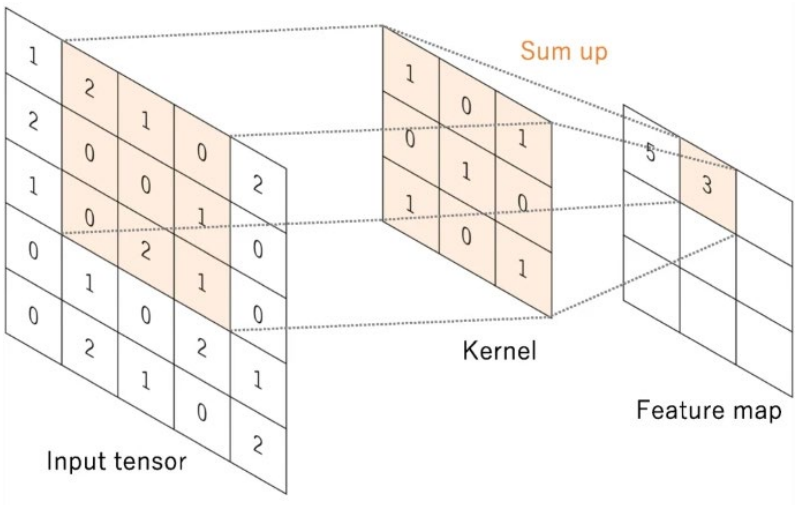
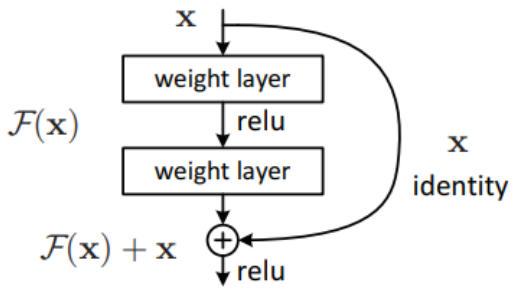


Figure 2. Visualization of a convolutional layer.
Source: Yamashita et al. (2018).

3. Residual Networks

One difficulty with developing large CNNs is that adding more layers does not simply increase the accuracy of the model. In fact, adding more layers could degrade the predictions due to the so-called "vanishing gradients" in backpropagation. To overcome this problem, He et al. (2015) developed what are known as Residual Networks (ResNet). They theorized that if one simply copied the output of a simpler model through the additional layers of a deeper model, the deeper model should never underperform the shallower one. Therefore, the ResNet architecture contains connections between layers, as seen in Figure 3, which allow the residual information from higher layers to be brought forward to subsequent layers. This allows very deep models to be created without degrading the output. Multiple ResNet architectures utilizing this technique are selected for the models developed in this work.



The identity mapping from an earlier layer, denoted by x, is shown being added to the output of a subsequent layer, $F(x)$ using a skip connection before the activation function, relu, is applied.

Figure 3. Visualization of a ResNet block. Source: He et al. (2015).

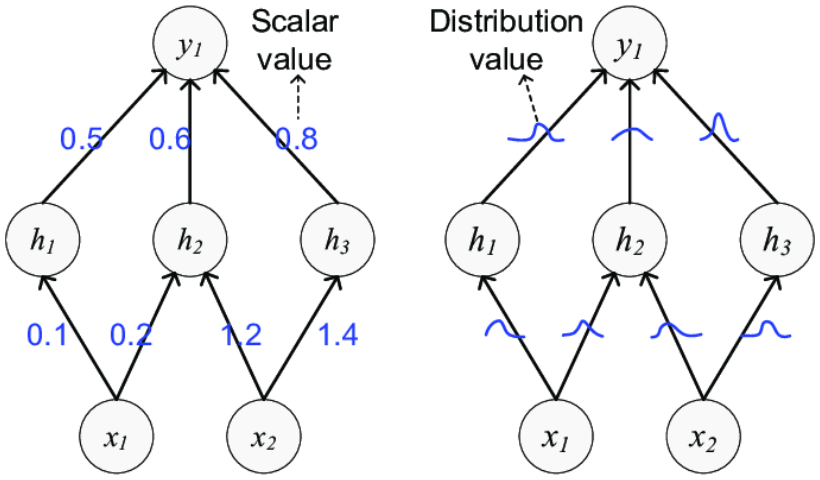
4. Bayesian Neural Networks

The final technique that will be discussed is BNNs, or in our case, Bayesian CNNs (BCNNs) which are simply BNNs that utilize a CNN architecture. BNNs combine the aspects of both neural networks as well as Bayesian probability theory to create models that can generate more accurate predictions and quantify the model's uncertainty in that prediction (Kendall and Gal 2017). Bayesian probability theory is built upon Bayes Theorem seen in Equation 1, which states that the probability of A occurring given a condition B is dependent upon the probability of A occurring multiplied by the probability of B occurring given A all divided by the probability of B. More simply, it describes how a prior probability can be updated given new information.

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)} \tag{1}$$

One can think of this as having a prior probability density function (PDF) which, when presented new evidence, shifts by some amount to form a new, posterior, probability density function. In a BNN, the model's weights and biases are represented by normal PDFs instead of point values; this is illustrated by Figure 4. The PDF is then updated using Bayes Theorem as the weights are trained. Now, when a BNN is used to make a prediction, different solutions can occur as random values are selected according to the PDF of each weight and bias. Using a Monte Carlo approach, one can run the input data through the

model many times to get a sense of the range of possible outcomes. Determining the deviation of these predictions can then be used to determine the model's uncertainty in that prediction.



The discrete weight values shown in traditional NN on the left compared to the distributions used in the BNN on the right.

Figure 4. Visual comparison of traditional and Bayesian NNs. Source: Jia et al. (2020).

BNNs are often implemented practically by using TensorFlow Probability, which has many of the features necessary to incorporate Bayesian statistics into a model. Specifically, two-dimensional convolutional layers and dense layers are replaced by Convolutional Flipout layers and Dense Flipout layers, respectively. These Flipout layers allow for the PDF weights characteristic of Bayesian models.

Since this model architecture deals with distributions rather than point values, a traditional loss function like Mean Squared Error cannot be used during training. Therefore, the goal when training the BNN is to maximize the evidence lower bound shown in Equation 2.

$$L_q = E_q[\log p(D|\mathbf{w})] - KL[q(\mathbf{w}) \| p(\mathbf{w})] \quad (2)$$

In Equation 2, the term D is a dataset, $p(\mathbf{w})$ is a prior on the weights, $q(\mathbf{w})$ is the posterior, so log $p(D|\mathbf{w})$ is the log likelihood, and KL is the Kullback-Leibler divergence (Kullback 1959). KL divergence is a technique to determine the similarity of two distributions and is utilized within the model during backpropagation. The negative log likelihood then serves as the loss function in the model. Ultimately, what this accomplishes is the same as with the simpler loss functions used with traditional NNs but adapted to be able to handle distributions rather discrete values.

These techniques allow BCNNs to produce uncertainty in predictions depending on the training dataset used and the feature vector the model is trying to predict; therefore, such models are no longer deterministic. As a result, a BNN model can function as an ensemble when multiple predictions are performed on the same dataset, enabling the creation of spatial uncertainty maps.

III. RAIN-TYPE CLASSIFICATION

Rain-type classification presents a simple case-study for the application of CNNs and BCNNs on spaceborne remote sensing data. In this chapter, a CNN and BCNN are trained from GMI PMW data to predict rain-type classification over the ocean.

A. DATA PROCESSING

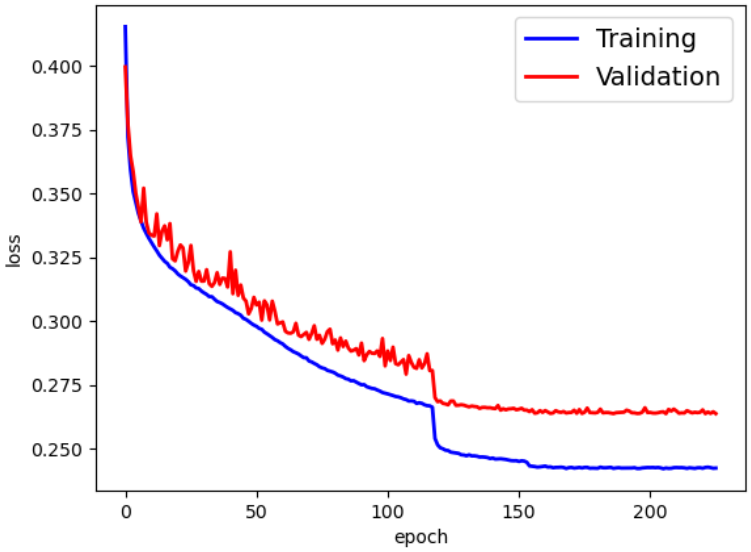
To train the rain-type classification models, GMI data collected over the ocean in 2017 from the GPM mission was used, containing around 10 million samples. GMI brightness temperatures from 13 microwave channels were selected and a 25 x 9 vector was extracted from each channel, which corresponds to a 100 km x 100 km field of view. The DPR onboard the CO contains a precipitation type flag indicating whether the radar observed convective or stratiform precipitation according to the algorithm described by Awaka et al. (2016). This flag was then assigned as the "true" label for matching GMI data points. Since both the GMI and DPR data were collected by GPM, the precise spatial and temporal matching of the two datasets was possible. An additional dataset was generated from data collected in 2018 and was used for evaluation and testing.

B. MODEL ARCHITECTURE AND TRAINING

Because the model needed to perform image recognition on the PMW brightness data, a convolutional neural network was selected. The ResNet (He et al. 2016) architecture has proven especially skilled at image classification using deep models and was chosen for this task. Specifically, a ResNetV1 with 32 ResNet blocks and 937,000 parameters was architecture used for training. A ResNetV2 model was also trained, but the most accurate results were achieved with V1. Different numbers of ResNet blocks were also attempted, but 32 layers was chosen after experimentation.

A Bayesian model was built off a traditional ResNetV1 architecture with 32 ResNet blocks but was adapted using the Flipout layers of TensorFlow Probabilities and the evidence lower bound shown in Equation 2, demonstrated in Zhu (2019), as the loss function to achieve a probabilistic output. To train the models, a base learning rate of 0.001

was used with the Adam optimizer. Learning rate was decreased if loss did not improve for 10 epochs in a row. The final model was achieved after around 200 epochs for both models after training on over 10 million samples. The learning curve for the final BCNN is shown in Figure 5. The abscissa represents the number times the model has trained through the full training dataset, also known as an epoch. The blue and red lines in the figure show the progression of the model's loss on the training and validation datasets, respectively. The significant drop in loss seen around epoch 120 corresponds to a reduction in the learning rate, allowing the model to become more accurate as it more precisely tuned the model weights.



The negative log likelihood loss values for the training (blue) and validation (red) datasets at the end of each epoch during training.

Figure 5. Plot of learning curves for the BCNN.

C. RESULTS

Both a Bayesian ResNet32 V1 and an equivalent traditional ResNet model were trained from the training dataset and achieved significant improvement over results reported in (Petković et al. 2019)—by more than 17%—when run on the test dataset. The implementation of a Bayesian approach shows improved accuracy, even when only running one Monte Carlo simulation, over classical ResNet32 V1 (Table 1). Increasing the

number of Monte Carlo simulations to 25 also leads to an additional 3% improvement in performance. The results of the final model's predictions are shown in Table 1.

Table 1. Model accuracy on test data

Architecture	***n* = 1**	***n* = 25**
Bayesian	87.7%	88.7%
Traditional	85.7%	-

The letter "*n*" denotes the number of Monte Carlo simulations utilized when making the prediction. Therefore, for traditional models *n*=1.

Additionally, the Bayesian model contains little bias towards stratiform or convective classifications in the test data, which has been a limitation of models developed in previous research (Henderson et al. 2017). This is shown in Table 2 in the form of a confusion matrix. The rows denote the actual event which occurred, and the columns represent what the model predicted. For example, 11.6% of stratiform observations were mislabeled as convective.

Table 2. Confusion matrix for the Bayesian model on test data

Stratiform	88.4%	11.6%
Convective	11.0%	89.0%
	Stratiform	Convective

The rows of the confusion matrix above denote what rain type was observed, while the columns represent what the model predicted.

The improvement in skill can also be seen when performing predictions on sample data. An example GPM swath was selected from over the South Pacific Ocean to demonstrate this. Figure 6 shows the non-Bayesian model's performance on the swath while Figure 7 shows the results from the Bayesian model. In each figure, "true" labels refer to classifications derived from DPR data. For both models, the general areas of

convection and stratiform precipitation are identified, although details are sometimes missed. For example, the areal coverage of convective cloud in the center of the swath (27°S 165°W) was overestimated by both models and the southernmost area of convective cloud (29°S 163°W) was predicted to extend farther east than was observed.

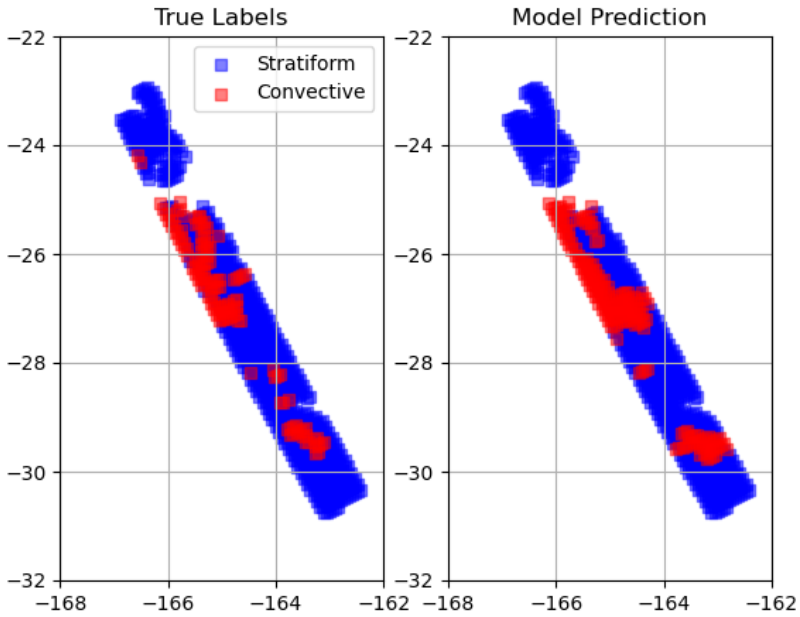


Figure 6. Example output from traditional model compared to true labels.

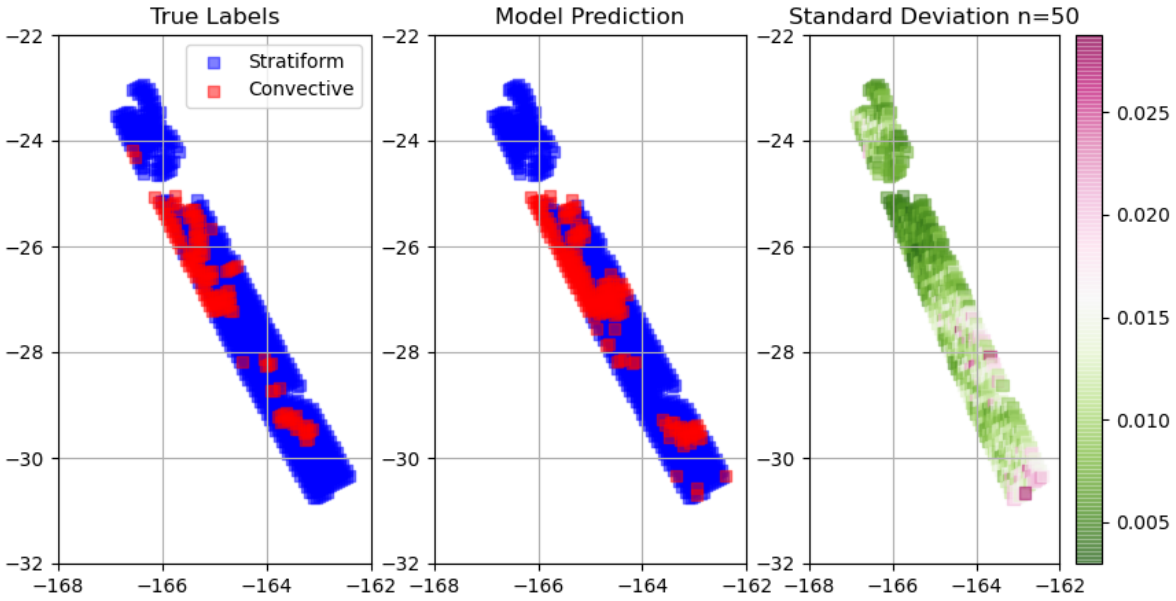


Figure 7. Example output from Bayesian model compared to true labels

Since the BCNN contains uncertainty in its prediction, this uncertainty can be plotted spatially, alongside the prediction. On the righthand side of the figure, the standard deviation of the model's prediction is shown, with higher uncertainty shown in purple. This uncertainty was calculated by taking the average standard deviation of the raw output from the model, known as the logits. The logits are the values output by the model before being transformed into a binary output through the use of the SoftMax function. This function simply takes the logits output by the model and selects the largest value to be set as the prediction.

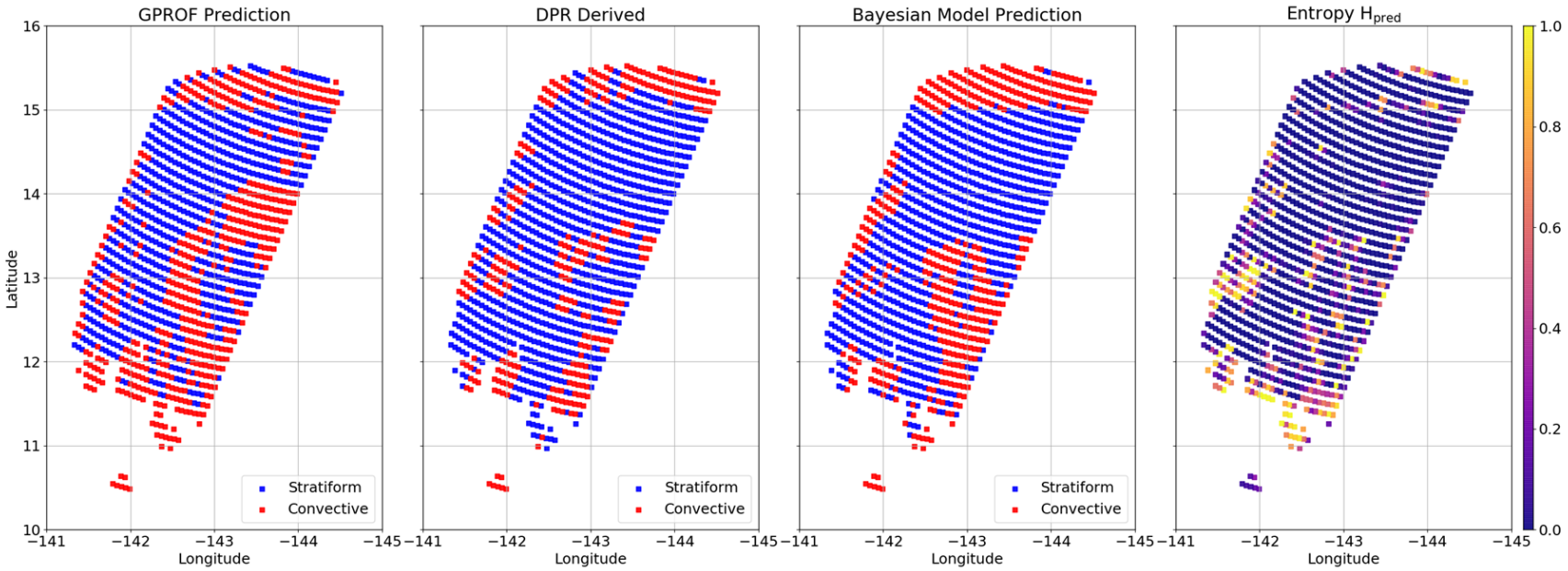
While not all incorrect classifications are uncertain, areas of uncertainty often have incorrect predictions in the vicinity, such as near 28°S 164°W, where multiple incorrect predictions occurred and a standard deviation of 0.015 was recorded. Similarly, at the southern end of the swath near 31°S 163°W where multiple incorrect predictions occurred in conjunction with a standard deviation of over 0.02.

Another sample swath was chosen to compare the BCNN's prediction and its uncertainty to the operational algorithm currently in use, GPROF V5. The results are plotted in Figure 8. In this example, uncertainty is calculated more accurately by using the binary output obtained after the SoftMax function is applied. To quantify the uncertainty of the predictions, a metric known as Entropy is used. The formula for Entropy is shown in Equation 3, where $H(X)$ is the entropy of a prediction, n is the number of classes, and $p(x_i)$ is the percentage of the predictions that match each class. Therefore, entropy is a means of showing the uncertainty of a prediction based on the distributions of the predictions among all the Monte Carlo simulations. If every Monte Carlo simulation predicts the same classification, the entropy will be zero; however, if predictions are evenly split between the two classes, entropy will equal one.

$$H(X) = -\sum_{i=1}^{n} p(x_i) \log p(x_i) \tag{3}$$

The BCNN ran the feature vectors through 200 Monte Carlo simulations to achieve the most accurate prediction and uncertainty values possible in this final example. When compared to DPR-generated labels and the Bayesian model, the operational prediction

from the GPROF algorithm seems to favor convective precipitation over stratiform, especially near 14°N 144°W. Furthermore, the highly uncertain areas, with entropy over 0.5, do seem to correlate to areas where the model performed more poorly (e.g., 13°N 142°W, 11°N 142.5°W). This also correlates to the edge of the precipitation which may be more difficult to predict because significant portions of the input vectors contain no precipitation.



Precipitation type classification results compared between the current operational scheme, GPROF V5 (left), the classification derived from the DPR which is used as training labels (center left), and the newly trained BCNN predictions (center right) after running the feature vectors through 200 Monte Carlo simulations. The entropy, or uncertainty of the BCNN results, (right) are plotted.

Figure 8. Bayesian model compared to GPROF V5 algorithm

THIS PAGE INTENTIONALLY LEFT BLANK

IV. REGRESSION

Regression presents a more difficult task when compared to classification. In this chapter, CNNs and BCNNs are trained from geostationary and GPM DPR data to predict radar reflectivity factor (in dBZ) and rainfall rates over the ocean near Central America. The predictions were conducted for a region where sea surface temperatures often exceed 28°C, and deep convection can form during any month. The moist, buoyant convection that forms in this region is dynamically similar to convection that forms in other warm ocean areas, such as the Western Pacific or Indian Ocean, so the model trained herein may possibly be extended for use in other areas of the world.

A. DATA PROCESSING

Data selection, matching, sampling, and normalizing were necessary to convert the satellite data into a format that NNs can successfully train from. GPM DPR data from March 2017 through July 2020 were used to generate the labels for the datasets used for training. Both the K_u-band radar reflectivity and estimated rainfall rate derived from radar were used as training labels for different models. The data were downloaded from the University of Washington GPM-K_u Data Set (University of Washington, 2020). Since, DPR radar data is three-dimensional, either a single altitude or composite reflectivity needed to be selected as the two-dimensional training input. Radar reflectivity factor from 1.5 km above the surface was used for training because it contained a large amount of high-quality data while still being relatively low in the atmosphere so that it corresponds closely with rain rate. The 1.5 km altitude also aligns well with where the lowest tilt angle of ground-based radars often scan at ranges of about 100 km. DPR data was restricted to the region between 20°N, 5°S, 95°W, and 60°W and excluded any areas over land. This was done to capture the marine convection over warm sea surface temperatures. This region is also near the GOES-16 subpoint and therefore has the data highest spatial resolution and the smallest amount of brightness temperature distortion due to zenith angle possible (Joyce et al. 2001).

After excluding all GPM data that did not fit the spatial selection criteria, the data was then matched to the associated GOES scans. GOES-16 completes a full-disk (FD) scan once every 10 or 15 minutes depending on the scanning mode of the satellite (Schmit et al. 2017). The FD scan requires 9 to 10 minutes to complete, and the area of the training data is scanned ~4 minutes after the start time of the scan. Since the GPM dataset only records a swath's start and end times, only the swath's maximum time offset with GOES-16 data is possible to calculate. GPM swaths that contained a maximum possible offset of ≤9 minutes were included in the dataset. This was to ensure that the GPM and GOES data matched as well as possible in time while also keeping enough data to train an effective model. Even with this consideration, small differences can arise during convection in the time between the GOES capture and GPM pass.

The GPM data was then co-located with the associated pixels in the FD scan. In December 2017, the GOES-16 subpoint was moved from 89°W to its operational location of 75°W. For each GPM/GOES pair, it was determined whether the observations were before or after this transition. Depending on the date, specific latitude/longitude to pixel conversions were used to account for the corresponding subpoint.

With the GPM dataset matched temporally and spatially, it was then resampled to balance the dataset before training. To do this, a total of 5,000 datapoints were selected per dBZ bin. The full and sampled dataset distributions can be seen in Figure 9. The blue bar graph denotes the original dataset before sampling while the orange bars show the distribution of the sampled dataset. The figure shows the unbalanced nature of the original dataset, with points between 15–30 dBZ contained nearly 100k samples per bin. However, bins over 45 dBZ contained less than 5,000 data points per bin and were therefore over-sampled by duplicating the datapoints within the bin a maximum of 50 times. This was done to create a more even distribution of samples. Without this under- and oversampling, the model would be incentivized to predict the most common values found in the dataset, since incorrectly predicting one of the rare values would contribute minimally to the total loss calculation. The final training dataset contained over 200,000 samples, not including samples used for validation. Additionally, 25 data points were taken from each bin between 20 and 50 dBZ, before sampling, to construct a test dataset.

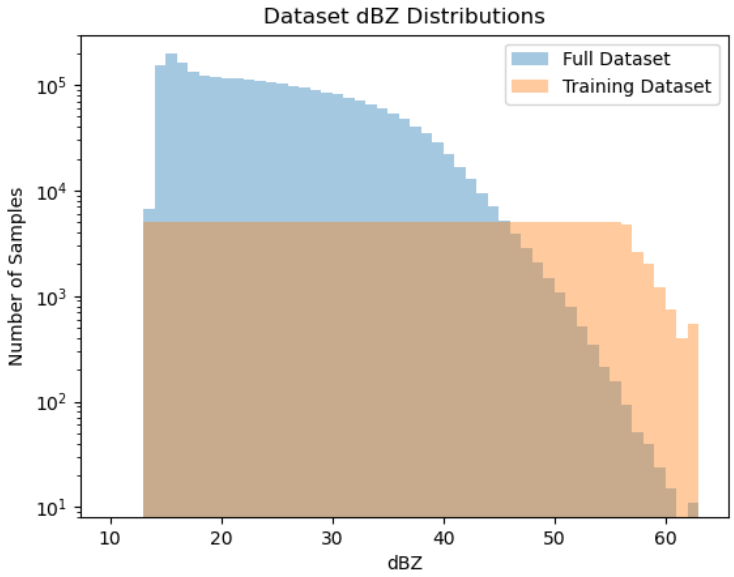


Figure 9. Distribution comparison between the full, unbalanced dataset and the dataset used for training

Most of the GPM data points contained no returns because there was no precipitation to detect. In this event, the dataset contained a fill value of -100 dB*Z* and was excluded from the training dataset. Any returns below the radar's effective sensitivity of 17 dB*Z*, which were often recorded in regions of no precipitation, were used to represent no precipitation.

The spatial distribution of the training dataset is shown in Figure 10. All points within a 1° x 1° area were counted from the training dataset then color filled based on the number of points it contained. Dark blue and green areas show locations where are large number of GPM observations were record. The figure shows that a large portion of the training data originated from the Intertropical Convergence Zone between 5°N and 10°N.

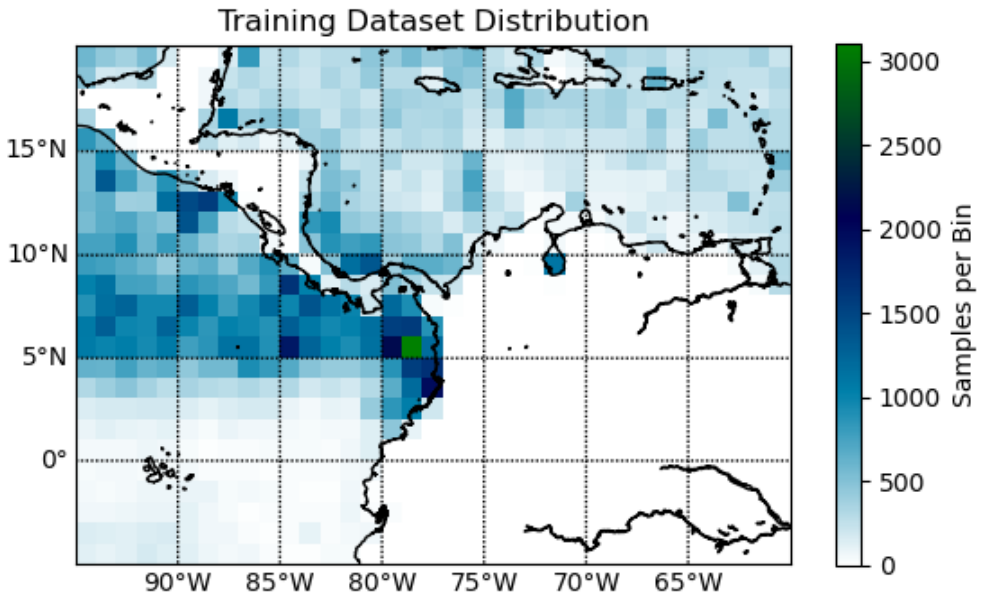


Figure 10. Spatial distribution of the datapoints used for model training

Finally, to create the training dataset, the GOES radiances from the nine IR bands captured by the ABI were extracted to function as the feature vector of the model. These bands were selected because the non-IR ABI bands, which detect reflection of solar radiation off of Earth, are only available during daytime, but the GPM data included both nighttime and daytime scenes. Multiple size "patches" were used to test the optimal size for the model. The sizes tested in this work were 9x9, 13x13, and 25x25 pixels, corresponding to 18km by 18km, 26km by 26km, and 50km by 50km areas, respectively. Figure 11 shows an example of one 25x25x9 pixel feature vector used in training, with each ABI band being displayed from band 8 on the far left to band 16 on the far right. The sample shown corresponded to a reflectivity of 51 dBZ, which was recorded by the DPR at the central pixel of the images within 9 minutes of the GOES image capture.

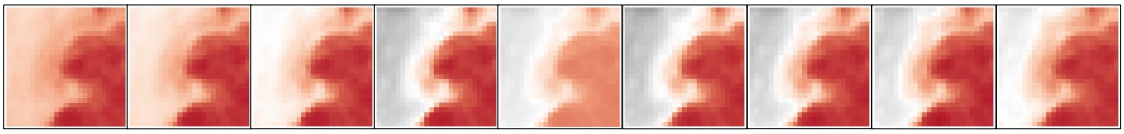


Figure 11. Example of a 25x25x9 feature vector used in training

B. MODEL ARCHITECTURE AND TRAINING

As in Chapter III, a ResNet architecture was utilized for all models trained. Both version 1 and version 2 ResNet20 models were used in. Larger architectures did not show noticeable improvement in loss, so were not tested further. Both 3x3 and 5x5 kernel sizes were tested; however, 3x3 achieved significantly lower loss values and was used with all models tested. The Adam optimizer was used during training as well as a batch size of 200 samples. Learning rate was the hyperparameter with the most impact on model training. Models training on rain rate achieved the lowest loss values with a base learning rate of 0.001 and models training on reflectivity achieved the lowest values with a learning rate of 0.0001. If validation loss did not improve for 10 epochs, learning rate was decreased to 10% of the previous learning rate. Feature vector augmentation was not performed in the final models due to the geospatial nature of the dataset. It was not known if rotation or flipping of the satellite data would negatively impact the model's skill due to a possible north-south or east-west relative relationship arising in the data that may be useful for prediction.

In total, six conventional models were trained for this experiment. A 9x9, 13x13, and 25x25 model were trained for predicting both reflectivity and rainfall rates. Only the 25x25 reflectivity predicting model reached lower loss values results from ResNet version 1, while the others all utilized ResNet version 2. Additionally, two BCNNs were trained to test the feasibility of using this technique to gauge uncertainty values from the precipitation regression. Both models were trained on the 9x9 patch dataset with the same hyper-parameters as the traditional models, however the negative log likelihood loss function was used during training rather than MSE.

C. RESULTS

The six models were tested on the test dataset, which contained data not seen in training, and MSE and MAE results are recorded in Table 3 and Table 4. Additionally, mean bias error (MBE), which determines the bias of the model by calculating the average error of each prediction without taking the absolute value first, is recorded in the two tables. MBE is positive when the average prediction is less than the true values, and negative when

greater. While results were similar between models, the 9x9 model performed most accurately when predicting reflectivity, while the 25x25 model performed most accurately when predicting rainfall rate.

Table 3. Results on reflectivity test data for the most accurate model for each vector size

	9x9	13x13	25x25
MSE (dBZ^2)	53	72	56
MAE (dBZ)	5.8	6.8	6.0
MBE (dBZ)	+1.6	+2.3	+1.8

Table 4. Results on rain rate test data for the most accurate model for each vector size

	9x9	13x13	25x25
MSE (mm^2 hr^{-2})	400	423	386
MAE (mm hr^{-1})	11.3	11.3	11.0
MBE (mm hr^{-1})	+3.0	+3.9	+3.4

The test results show the models' accuracy on the test dataset, which is composed of 700 samples between 20 to 50 dBZ not included in the training set. The models trained to predict reflectivity all have a MAE of around 6 dBZ which is small enough that they are capturing the general picture of the precipitation, though are not depicting any of the smaller scale features contained within the storms. Additionally, the MSE values, which weigh larger errors more heavily, are around 55 dBZ^2. MSE values in this range correspond to errors closer to 7.5 dBZ, which indicates there are outliers among the predictions that have larger errors still. For comparison, operational radars are typically calibrated to within 1 dBZ of a reference. The results from the MBE calculation show that all three models

underpredicted the reflectivity slightly more than they overpredicted it. The 9x9 model had the smallest bias: +1.6 dB*Z* on average.

The rainfall rate results show more significant outliers when compared to the previous data. The MAE values around 11 mm hr^{-1}; however, the MSE values correspond to errors of around 20 mm hr^{-1}. According to the MBE, the three models also underpredicted rainfall rate on average by 3–4 mm hr^{-1}. These values are likely due to the larger range of possible values, between 0 and 300 mm hr^{-1}, which was not encountered when predicting reflectivity due to being scaled logarithmically.

Predictions on GPM swaths not included in training or testing were also performed. The results of one swath for both reflectivity and rain rate prediction can be seen in Figure 12 and Figure 13. These plots compare the predictions from the three models along with the actual values observed by the DPR during the orbit. This data is displayed on top of the accompanying GOES-16 10.3-micron band to show how the prediction compares to the spatial distribution of infrared radiances. This provides a qualitative look at the performance of each model.

As with the test dataset, the 9x9 model performed slightly more accurately than the other two models when predicting dB*Z* values. All three models struggled with overpredicting 20–30 dB*Z* echoes located away from convective cells and predicted the size of convective cores to be larger than what was observed. However, the location of and maximum dB*Z* within the strongest convective cells were generally captured. Looking specifically at the 9x9 model, all the major convective cells were correctly placed, including the northwest to southeast orientation of the precipitation near 6.5°N 83.5°W, for example. The maximum reflectivity values in each cell were correctly predicted within 5 dB*Z* except for somewhat overestimating the previously mentioned cell and slightly underestimating reflectivity in the cell near 3°N 84.5°W.

For predicting rain fall rates, the 25x25 patch model slightly outperformed the other two models quantitatively, but the 9x9 model appeared to capture the isolated nature of the

heavy rain rate cores more accurately when looking qualitatively. All three models had problems predicting 5–25 mm hr^{-1} rain rates in large areas around the main convection associated with colder cloud tops, although the extent of this region is less than that found with the reflectivity models. Despite ultimately recording a larger error than the 25x25 model, the 9x9 model appeared to correctly identify the maximum magnitude of rain rate in the most intense convective cores. This may be because the 25x25 model produce more "average" rainfall predictions that produce less error overall, but a less realistic looking result. For example, the 9x9 model correctly predicted the isolated cores with rain rates near 75 mm hr^{-1} in the storms near 3°N 84.5°W, 3N 85.5°W, and the storm just south of 8°N 83°W, while also correctly predicting no cores with rain rate exceeding 75 mm hr^{-1} in the storms near 4°N 86°W, and 6.5°N 84.5°W. In contrast, the 25x25 model did not predict 75 mm hr^{-1} rainfall rates in any of the storms.

As with the traditional models, the performance of the BCNNs were determined by calculating statistics on the test dataset. The results for these tests can be seen in Table 5 and Table 6. The value "*n*" denotes the number of Monte Carlo simulations ran for the prediction. These test results show slightly below average performance when compared to the deterministic models with only a minor accuracy increase with increased Monte Carlo runs. The MBE also aligns with the non-Bayesian models, slightly under predicting both reflectivity and rainfall rate on average. However, these models were trained with the same hyperparameters used for the non-Bayesian models for simpler comparison, therefore performance improvements may be possible through further hyperparameter searching.

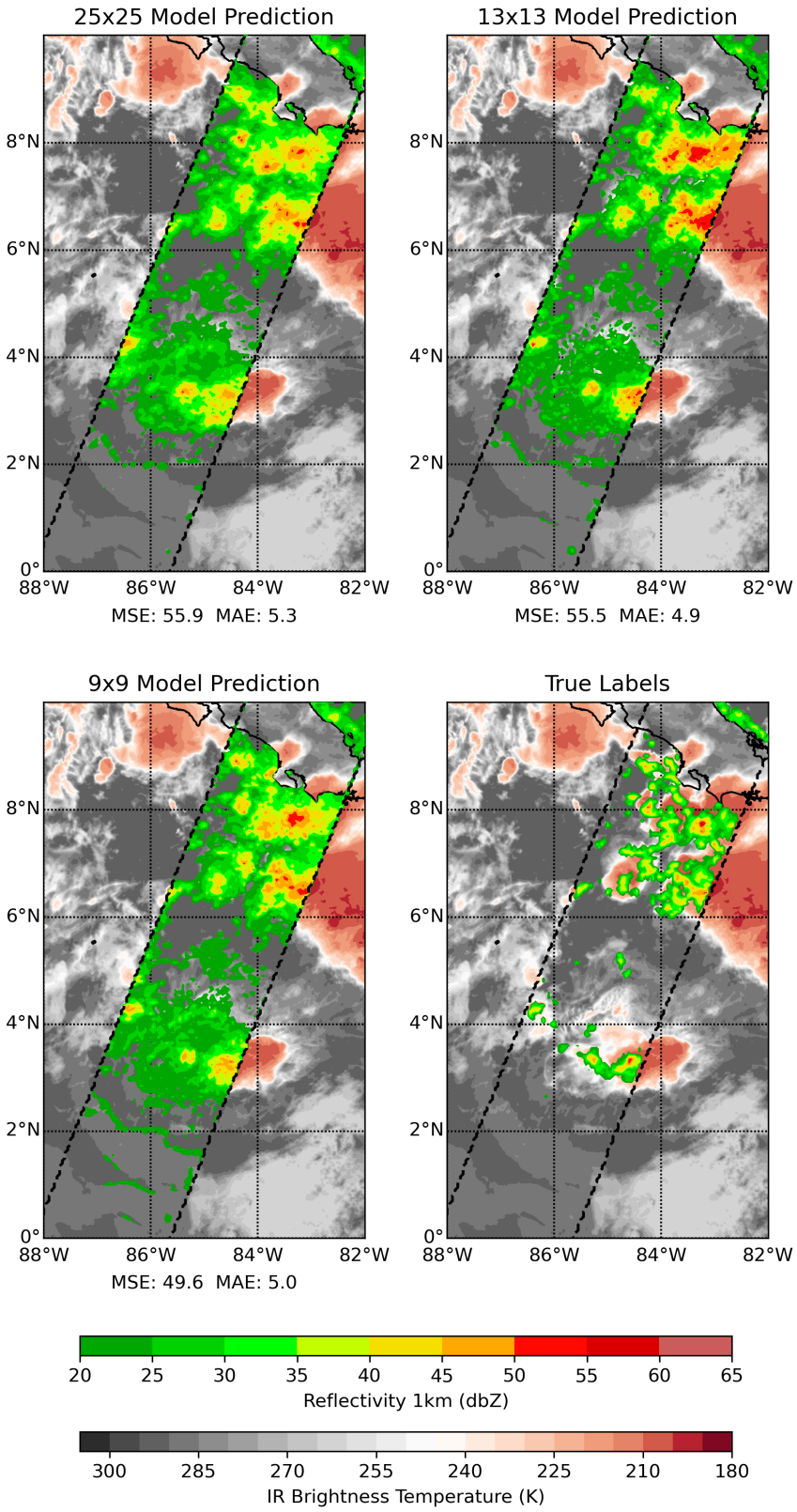


Figure 12. Model predictions compared to the true reflectivity observations

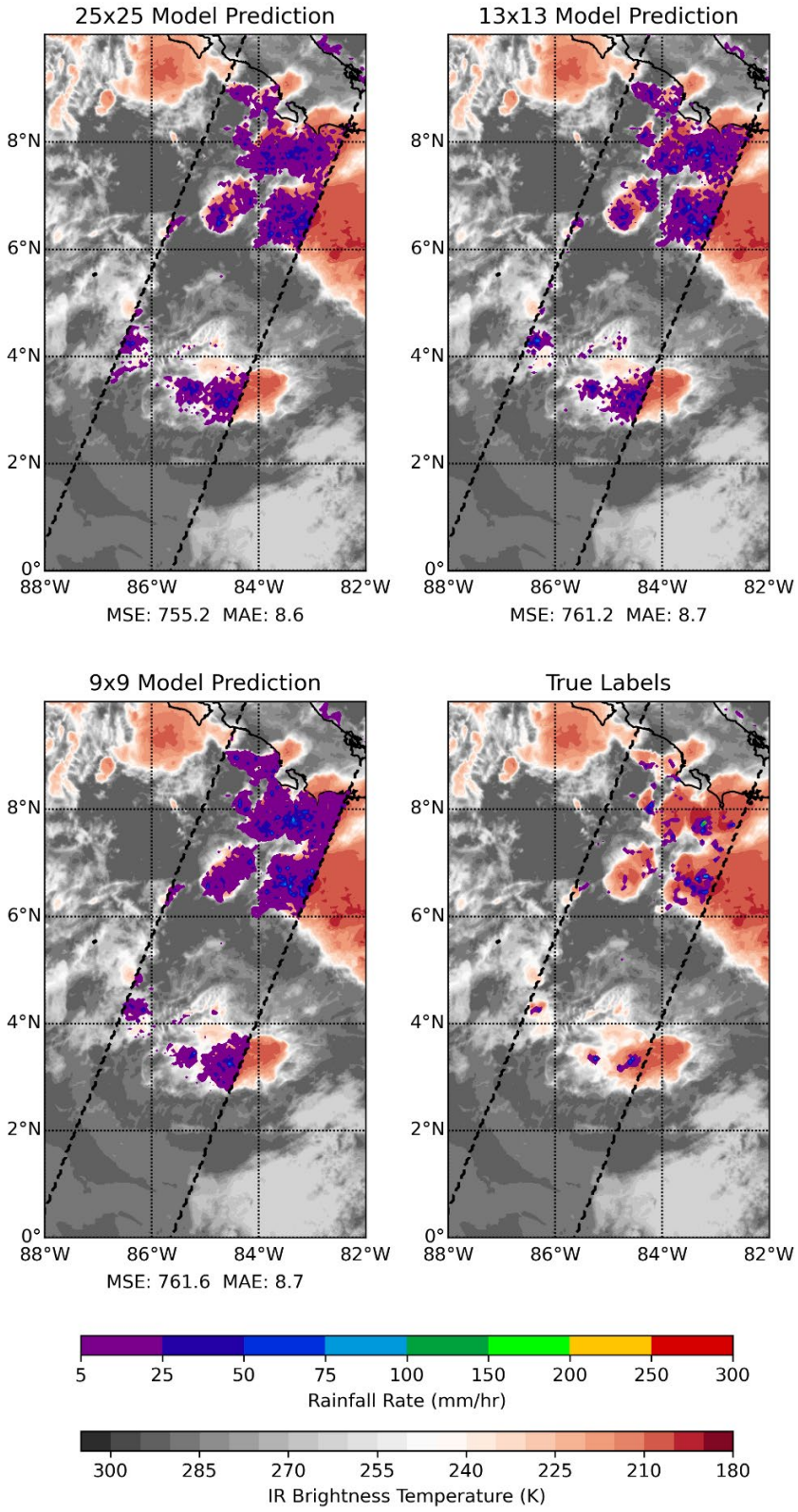


Figure 13. Model predictions compared to GPM estimated rainfall rate

Table 5. Test results for reflectivity prediction by the BCNN

	$n = 1$	$n = 25$
MSE (dBZ^2)	58	57
MAE (dBZ)	6.0	6.0
MBE (dBZ)	+1.7	+1.7

Table 6. Test results for rain rate prediction by the BCNN

	$n = 1$	$n = 25$
MSE (mm^2 hr^{-2})	460	451
MAE (mm hr^{-1})	12.6	12.5
MBE (mm hr^{-1})	+4.1	+4.0

Additionally, sample predictions were generated with the BCNN models. Figure 14 and Figure 15 show the results of the reflectivity and rain rate predictions, respectively, after running 100 Monte Carlo simulations. The final predictions are determined by taking the average of all 100 predictions. Like the test data, the models performed slightly less accurately than the traditional models in this scenario; however, additional uncertainty information is available with these models and is displayed in the left subplot of each figure. This uncertainty was calculated by determining the standard deviation of each prediction across the 100 runs for each point of the sample swath.

The uncertainty values are largest in areas where predicted rainfall is largest. The maximum standard deviation seen in the reflectivity prediction is less than 3 dBZ, which is much lower than the 6 dBZ MAE calculated. The maximum standard deviation for rainfall is less than the calculated MAE as well, at around 7 mm hr^{-1}. Ideally, the uncertainty range should encompass the errors of the model. If this is the case, the user of the model can likely trust that after considering both the prediction and its uncertainty, the true value is likely within that range. The cause of these low values may be that the size of the dataset is too small, leading to overfitting. In this case, the model may begin to

memorize the training dataset, leading to the model being highly certain in everything it predicts, despite being incorrect. Additional Bayesian architectures should be tested in future work to determine if this is possibly a result of the architecture used as well.

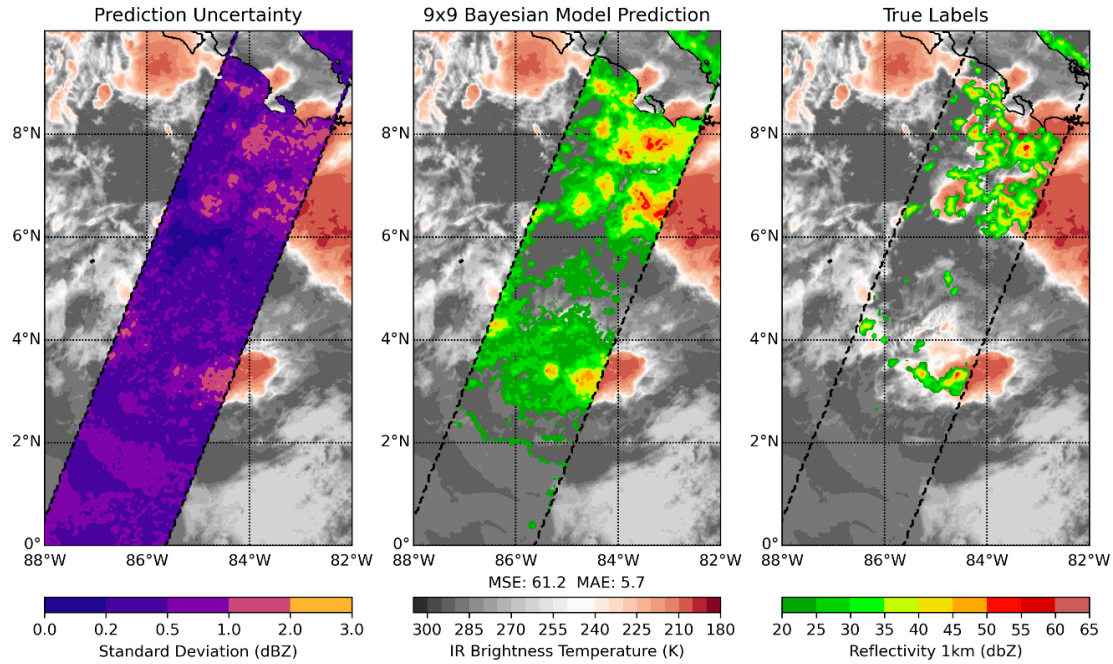


Figure 14. BCNN reflectivity prediction results

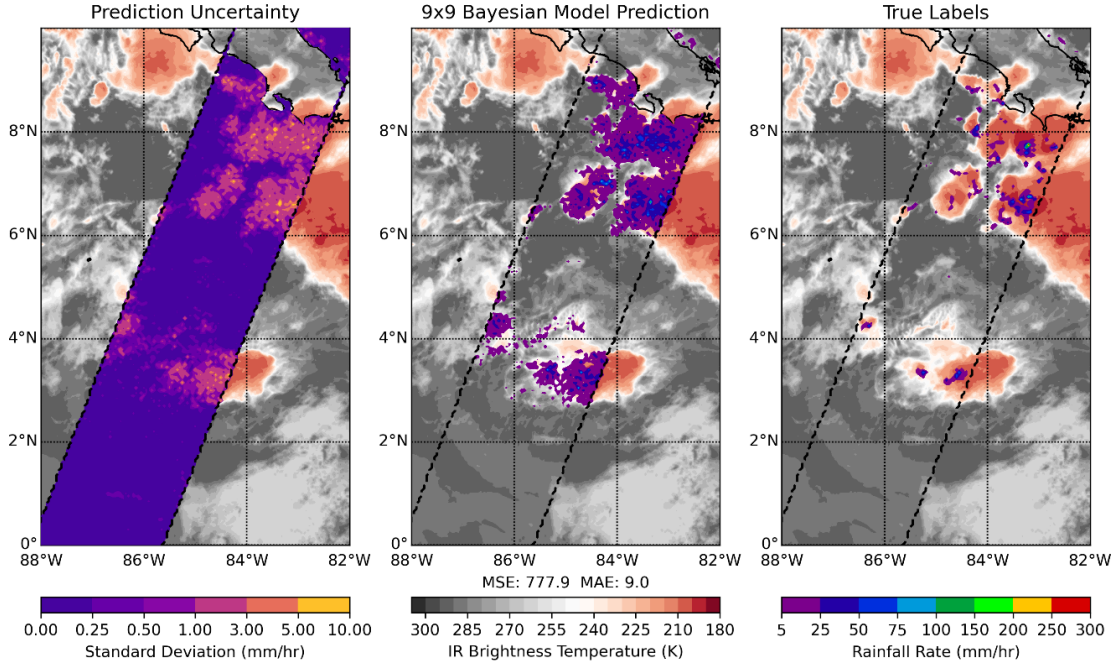


Figure 15. BCNN rain rate prediction results

V. CONCLUSION

The application of CNNs on the growing GPM dataset provides many opportunities to advance research and operational knowledge of global precipitation. This work demonstrated two possible applications of CNNs on this data: rain type classification and precipitation regression. These two experiments showed the possible advantages and difficulties of training both types of models on the GPM dataset.

The classification models developed in Chapter III showcased the ability of a CNN and BCNN to determine whether GMI radiances represented stratiform or convective precipitation. These models were trained by using a high quality, balanced dataset with GMI observations as feature vectors and DPR-derived precipitation type as labels. The traditional CNN reached an accuracy of 85.7% on test data while the Bayesian model achieved 87.7%, improving over previous literature (e.g., Petković et al. 2019).

The results of the classification BCNN not only shows the usefulness of BCNNs in rain type classification, but in atmospheric and geospatial data in general. Research into more advanced BCNN architectures could enhance this capability further, improving classification in similar applications—for example, in automatically identifying cloud or sea ice from multispectral satellite imagery with classification uncertainties attached to all predictions. Thus, expanded applications of the uncertainty information provided by BCNNs could prove useful in research and operational decision making.

Chapter IV demonstrated the possibility of using GPM DPR data to train a CNN to predict reflectivity and rain rate from GOES-16 data. Patches of nine GOES bands were used as feature vectors and DPR reflectivity and derived rainfall rate were used as training labels. The results achieved are promising for a first attempt given the small sample size employed, and they demonstrate a proof of concept in using these two data sources to produce a skillful model. Evaluation on test data achieved an MSE and MAE of 53 dBZ^2 and 5.8 dBZ respectively for the most accurate reflectivity model and 386 mm^2 hr^{-2} and 11 mm hr^{-1} for the most accurate rain rate model. Both reflectivity and rainfall rates were slightly under predicted when ran on test data, which was comprised of samples between

20 and 50 dB*Z*. When performing predictions on sample swaths not included in training, the models were able to accurately predict where the cores of the storms occurred and often also the maximum dB*Z* value observed. However, the models trained overestimated the extent of the strongest precipitation and greatly overestimated the areal coverage of echoes between 20–30 dB*Z*. Similarly, the rain rate models usually overestimated the extent of the regions of 5 mm hr^{-1} rain. They were often able to identify which storms did and did not contain heavy rain cores in addition to predicting the approximate locations of the heaviest rain. However, the model still underestimated the highest rain rates found in some cores. Because such rain rates are relatively rare, a larger dataset may help improve model performance on the most intense radar echoes and rain rate predictions. The hope of using a CNN as the model architecture was to allow for the model to independently distinguish the differing patterns in the GOES patches when comparing patches of similar brightness temperature. One example would be distinguishing between cold cirrus clouds not associated with active precipitation, and similarly cold areas where active convection is occurring. While this goal was not completely achieved, the models were able to avoid putting significant deep convection in areas of non-precipitating anvil clouds, which shows progress towards this ultimate goal.

Additionally, the BCNNs trained on the precipitation datasets demonstrated the feasibility of this technique on generating two-dimensional fields that are traditionally derived from radar data and quantifying uncertainties in the predictions using geostationary radiances only. The slightly decreased performance of the BCNN relative to non-Bayesian models may be due to using the same hyperparameters as the non-Bayesian models despite utilizing a different model architecture. Additionally, the increased number of trainable parameters in a BCNN model when compared to a traditional CNN may have led to increased overfitting. With nearly double the trainable parameters, the BCNN may not generalize as well as the smaller, non-Bayesian models when training off a smaller dataset, such as with the one used for precipitation regression. The smaller dataset size may also explain the low uncertainty values found with the BCNN predictions. If the model is overfitting to the dataset, this may cause the model to create predictions which are highly

certain, despite being incorrect. Additional work would need to be done to determine if this is the case.

While the architectures and training techniques used to train the regression models were similar to in Chapter III, the datasets were very different. The differing results between the two experiments was most likely caused by the different sizes of the datasets used for training, in addition to the increased difficulty found inherently with regression tasks. There are three qualities of the GPM and GOES datasets that make training a highly accurate model from them more difficult and which could be addressed in future work.

The first challenge is that the K_u-Band radar onboard the DPR only has an effective sensitivity down to 17 dB*Z*, with significant noise below this value down to around 12 dB*Z*. The rest of the dataset does not contain information about the reflectivity because scatterers were not present everywhere. This discontinuity and noise in the data leads to the decision of whether to include these values in the training set entirely, and if so how to incorporate them. A possible solution would be to design a classification model to determine whether there was rain or no rain, before using a regression model to then predict the actual value of the points classified as "rain." It was decided to go the simpler route in this work and simply not include the "no rain" values in the training, however with this decision, a significant portion of the dataset was never used in training. Including this data may have allowed for more skill in predicting areas of no returns in regions such as anvil clouds, which was a limitation of the models trained in this work.

The second difficulty with the dataset is that, like many in geosciences, it is highly unbalanced. There are four orders of magnitude more 20 dB*Z* observations than there are 60 dB*Z* observations, even though the 60 dB*Z* observations are equally--if not more—important because they correspond with the most extreme weather events. The distribution of rain rate observations is even more unbalanced, with hundreds of thousands of observations below 5 mm hr^{-1} while only a few thousand exist between 100 and 300 mm hr^{-1}. However, the rain rate distribution does avoid the discontinuity found in the reflectivity data because where no rain is present, the rain rate is zero. Under- and oversampling was used to correct this imbalance in the final dataset used for model training, however this greatly reduced the size of the final training dataset while also

possibly inducing overfitting due to extensive duplication of strong echoes. The most effective means to fix this issue is the creation of a much larger dataset. This could be done by including other satellite data that observes convection similar to that studied herein. This could include GOES-16 data from a larger area and longer time, as well as GOES-17 or Himawari-8 data. Other techniques such as a weighted loss function, which considers the rarity of the sample when calculating the loss, could provide more accurate results without having to increase the full dataset's size. An ideal solution may be to do both.

Finally, an additional challenge faced with creating a training dataset between GPM and geostationary data is temporally matching the observations. Because the GPM dataset only records the start and end time of each swath, the actual observation time of each datapoint is not known. While a maximum time offset of 9 minutes was determined for the training dataset, 9 minutes may still be too large of a time differential allowable to train a truly accurate model. Tropical convection can often develop quickly and with very sharp gradients between heavy precipitation and no precipitation, leading to labels being offset from where the storm is located on the GOES scan. Incorrect labeling significantly hinders the model's ability to learn the fine scale patterns necessary to distinguish smaller scale storm structure. Further constraining of the maximum time offset would alleviate this problem, however it would reduce the amount of data useable for training, causing the imbalance in the dataset to increase. A larger dataset and/or a better understanding of the time of each DPR observation would be beneficial in alleviating this possible source of error.

To conclude, both experiments performed in this thesis have highlighted possible applications of NNs on GPM data with successful results. The rain type classification model outperformed pervious research and operational models while also demonstrating the successful application of a BCNN with improved results. The precipitation regression models developed in Chapter IV showcases a successful attempt at building both conventional and Bayesian models from the combination of both GPM and GOES datasets while also highlighting the likely steps required to train more accurate models in future work. These results, along with research by others in the field show that the combination of CNNs and geospatial datasets such as those produced by GPM will continue to advance

operational and research objectives within and beyond the atmospheric sciences community as technologies and knowledge advances with further work.

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

Anagnostou, E. N., 2004: A convective/stratiform precipitation classification algorithm for volume scanning weather radar observations. *Meteor. Appl*., **11**, 291–300, https://doi.org/10.1017/S1350482704001409.

Arkin, P. A., and P. E. Ardanuy, 1989: Estimating climatic-scale precipitation from space: A review. *J. of Climate*, **2**, 1229–1238, https://doi.org/10.1175/1520-0442(1989)002<1229:ECSPFS>2.0.CO;2.

Awaka, J., M. Le, V. Chandrasekar, N. Yoshida, T. Higashiuwatoko, T. Kubota, and T. Iguchi, 2016: Rain type classification algorithmmodule for GPM dual-frequency precipitation radar. *J. Atmos Oceanic Technol*., **33**, 1887–1898, https://doi.org/10.1175/JTECH-D-16-0016.1.

Behrangi, A., K. Hsu, B. Imam, S. Sorooshian, G. J. Huffman, and R. J. Kuligowski, 2009: PERSIANN-MSA: A precipitation estimation method from satellite-based multispectral analysis. *J. Hydrometeor*., **10**, 1414–1429, https://doi.org/10.1175/2009JHM1139.1.

Clayson, C. A., J. Edson, A. Paget, R. Graham, and B. Greenwood, 2019: Effects of rainfall on the atmosphere and the ocean during SPURS-2. *Oceanogr*., **32**, 86–97, https://doi.org/10.5670/oceanog.2019.216.

Glorot, X., A. Bordes, and Y. Bengio,Year 2011: Deep sparse rectifier neural networks. *Vol. 15 of Proceedings of Machine Learning Research*, Proceedings of the 14th International Conference on Artificial Intelligence and Statistics, Fort Lauderdale, FL, J. of Machine Learning Research; 315–323 http://proceedings.mlr.press/v15/glorot11a.html.

Goodfellow, I., Y. Bengio, and A. Courville, 2016: *Deep Learning*. MIT Press, http://www.deeplearningbook.org.

Hagos, S., Z. Feng, R. S. Plant, and A. Protat, 2020: A machine learning assisted development of a model for the populations of convective and stratiform clouds. *J. Adv. Model. Earth Sys*., **12**, https://doi.org/10.1029/2019MS001798.

He, K., X. Zhang, S. Ren, and J. Sun; 2015; Deep residual learning for image recognition; *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*; 770–778; arXiv:1512.03385.

Henderson, D. S., C. D. Kummerow, D. A. Marks, and W. Berg, 2017: A regime-based evaluation of TRMM oceanic precipitation biases. *J. Atmos. Oceanic Technol*., **34**, 2613–2635, https://doi.org/10.1175/JTECH-D-16-0244.1.

Hilburn, K. A., I. Ebert-Uphoff, and S. D. Miller, 2021: development and interpretation of a neural-network-based synthetic radar reflectivity estimator using GOES-R satellite observations. *J. Appl. Meteor. and Climatol*., **60**, 3–21, https://doi.org/10.1175/JAMC-D-20-0084.1.

Hou, A. Y., Kakar, R. K., Neeck, S., Azarbarzin, A. A., Kummerow, C. D., Kojima, M., Oki, R., Nakamura, K., & Iguchi, T, 2014: The global precipitation measurement mission. *Bull. Amer. Meteor. Soc.*, **95**, 701–722. https://doi.org/10.1175/BAMS-D-13-00164.1.

Houze, R. A., Jr., 1993: *Cloud Dynamics*. Academic Press, 573 pp.

Huffman, G., R. Adler, M. Morrissey, D. Bolvin, S. Curtis, R. Joyce, B. McGavock, and J. Susskind, 2001: Global precipitation at one-degree daily resolution from multisatellite observations. *J. Hydrometeor*., **2**, https://doi.org/10.1175/1525-7541(2001)002<0036:GPAODD>2.0.CO;2.

Huffman, G. J., and Coauthors, 2007: The TRMM multisatellite precipitation analysis (TMPA): Quasi-global, multiyear, combined-sensor precipitation estimates at fine scales. *J. of Hydrometeor*., **8**, 38–55, https://doi.org/10.1175/JHM560.1.

Iguchi, T., S. Seto, R. Meneghini, N. Yoshida, J. Awaka, M. Le, V. Chandrasekar, and T. Kubota, 2017: GPM/DPR level-2. Algorithm Theoretical Basis Doc., 81 pp., http://www.eorc.jaxa.jp/GPM/doc/algorithm/ATBD_DPR_201708_whole_1.pdf.

Jia, X., J. Yang, R. Liu, X. Wang, S. Cotofana, and W. Zhao, 2020: Efficient computation reduction in Bayesian neural networks through feature decomposition and memorization. *IEEE Trans. Neural Netw. Learn. Syst.*, **PP**, 1–10, https://doi.org/10.1109/TNNLS.2020.2987760.

Joyce, R., J. Janowiak, and G. Huffman, 2001: Latitudinally and seasonally dependent zenith-angle corrections for geostationary satellite IR brightness temperatures. *J. Appl. Meteor*., **40**, 689–703.

Joyce, R. J., and P. Xie, 2011: Kalman filter–based CMORPH. *J. Hydrometeor*., **12**, 1547–1563, https://doi.org/10.1175/JHM-D-11-022.1.

Kendall, A., and Y. Gal, 2017: What uncertainties do we need in Bayesian deep learning for computer vision? *Proceedings of the 31st International Conference on Neural Information Processing Systems*, Red Hook, NY, USA; Curran Associates Inc., 5580–5590.

Krizhevsky, A., I. Sutskever, and G. E. Hinton, 2012: ImageNet classification with deep convolutional neural networks. *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'12, Red Hook, NY, USA, Curran Associates Inc., 1097–1105 (Accessed February 9, 2021).

Kullback, S., 1959: *Information Theory and Statistics*. Wiley, 395 pp.

Kummerow, C., and Coauthors, 2000: The status of the tropical rainfall measuring mission (TRMM) after two years in orbit. *J. Appl. Meteor.*, **39**, 1965–1982, https://doi.org/10.1175/1520-0450(2001)040<1965:TSOTTR>2.0.CO;2.

Kummerow, C. D., D. L. Randel, M. Kulie, N.-Y. Wang, R. Ferraro, S. J. Munchak, and V. Petkovic, 2015: The evolution of the Goddard profiling algorithm to a fully parametric scheme. *J. Atmos. Oceanic Technol.*, **32**, 2265–2280, https://doi.org/10.1175/JTECH-D-15-0039.1.

NASA, 2020: Global precipitation measurement mission. Accessed 10 February 2021, https://gpm.nasa.gov/missions/GPM.

Oh, K.-S., and K. Jung, 2004: GPU implementation of neural networks. *Pattern Recognit.*, **37**, 1311–1314, https://doi.org/10.1016/j.patcog.2004.01.013.

Okazaki, A., T. Honda, S. Kotsuki, M. Yamaji, T. Kubota, R. Oki, T. Iguchi, and T. Miyoshi, 2019: Simulating precipitation radar observations from a geostationary satellite. *Atmos. Meas. Tech.*, **12**, 3985–3996, https://doi.org/10.5194/amt-12-3985-2019.

Petković, V., M. Orescanin, P. Kirstetter, C. Kummerow, and R. Ferraro, 2019: Enhancing PMW satellite precipitation estimation: detecting convective class. *J. Atmos. Oceanic Technol.*, **36**, 2349–2363, https://doi.org/10.1175/JTECH-D-19-0008.1.

Powell, S. W., R. A. Houze, and S. R. Brodzik, 2016: Rainfall-type categorization of radar echoes using polar coordinate reflectivity data. *J. Atmos. Oceanic Technol*, **33**, 523–538, https://doi.org/10.1175/JTECH-D-15-0135.1.

Randel, D. L., C. D. Kummerow, and S. Ringerud, 2020: The Goddard profiling (GPROF) precipitation retrieval algorithm. *Satellite Precipitation Measurement: Volume 1*, V. Levizzani, C. Kidd, D.B. Kirschbaum, C.D. Kummerow, K. Nakamura, and F.J. Turk, Eds., Advances in Global Change Research, Springer International Publishing, 141–152.

Schmit, T. J., P. Griffith, M. M. Gunshor, J. M. Daniels, S. J. Goodman, and W. J. Lebair, 2017: A closer look at the ABI on the GOES-R Series. *Bull. Amer. Meteor. Soc.*, 98, 681–698, https://doi.org/10.1175/BAMS-D-15-00230.1.

Schumacher, C., R. A. Houze, and I. Kraucunas, 2004: The tropical dynamical response to latent heating estimates derived from the TRMM precipitation radar. *J. Atmos. Sci.*, **61**, 1341–1358, https://doi.org/10.1175/1520-0469(2004)061<1341:TTDRTL>2.0.CO;2.

Steiner, M., and R. A. Houze, Jr., 1997: Sensitivity of the estimated monthly convective rain fraction to the choice of Z–R relation. *J. Appl. Meteor.*, **36**, 452–462, https://doi.org/10.1175/1520-0450(1997)036<0452:SOTEMC>2.0.CO;2.

Takayabu, Y. N., and W.-K. Tao, 2020: Latent heating retrievals from satellite observations. *Satellite Precipitation Measurement: Volume 2*, V. Levizzani, C. Kidd, D.B. Kirschbaum, C.D. Kummerow, K. Nakamura, and F.J. Turk, Eds., Advances in Global Change Research, Springer International Publishing, 897–915.

Thompson, E. J., S. A. Rutledge, B. Dolan, M. Thurai, and V. Chandrasekar, 2018: Dual-polarization radar rainfall estimation over tropical oceans. *J of Appl. Meteor. Climatol.*, **57**, 755–775, https://doi.org/10.1175/JAMC-D-17-0160.1.

University of Washington, 2020: GPM-Ku Data Set, Version 5. Subset: South America, access date January 5, 2021, http://gpm.atmos.washington.edu/.

Veillette, M. S., E. P. Hassey, C. J. Mattioli, H. Iskenderian, and P. M. Lamey, 2018: Creating synthetic radar imagery using convolutional neural networks. *J. Atmos. Oceanic Technol.*, **35**, 2323–2338, https://doi.org/10.1175/JTECH-D-18-0010.1.

Wang, Y., L. Tang, P.-L. Chang, and Y.-S. Tang, 2021: Separation of convective and stratiform precipitation using polarimetric radar data with a support vector machine method. *Atmos. Meas. Tech.*, **14**, 185–197, https://doi.org/10.5194/amt-14-185-2021.

Yamashita, R., M. Nishio, R. K. G. Do, and K. Togashi, 2018: Convolutional neural networks: an overview and application in radiology. *Insights Imaging*, **9**, 611–629, https://doi.org/10.1007/s13244-018-0639-9.

Zhu, Lingchen, 2019: TensorFlow Probability tutorial. Accessed 10 February 2021, https://github.com/zhulingchen/tfp-tutorial

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California