Using Deep Convolutional Generative Adversarial Neural Networks to Predict Precipitation Data

By the Weather Project Team- Ambiorix Cruz Angeles, Satyam Sharma, Airton Prado, Yuan Zhao, and Sami Abood

1. Abstract

Weather forecasting is a canonical predictive challenge that has depended primarily on model-based methods. In this paper we study different algorithm models used by my team to predict rainfall via different approaches like using conditional GAN and simpler models like linear regression. As being part of the evaluation team of the group this paper will mostly focus on creating a baseline using a linear regression model.

The baseline was used to compare the other various models that were created by other group members. In addition the paper will talk about the three different methods MSE, SSIM and confusion matrix that were used in order to evaluate the results and get a numerical assessment.

2. Introduction

Weather radar has advanced greatly and has played important roles for meteorological and climatological applications. It has the ability to detect and warn hazards associated with severe local storms, hail, tornadoes, high winds and intense precipitation. Radar works by sending out particles that will be reflected and then collecting the reflected particles to create an image.

For weather like rain the particles would be reflected off of precipitation with the density depending o the amount of particles being reflected.

Unfortunately, radar images will not always reflect what is occurring in the atmosphere and not everything that it shows will be precipitation. For example, the radar could detect precipitation that will not reach the ground. In addition if the radar is close to the coast and the beam is broad enough, it may reflect off the sea and return strong reflectivity that's considered sea clutter and not rain.

The research being done is important because weather radar is a great source for precipitation and not being able to have them over the ocean leaves us blind and unprepared on a big area of the Earths surface.

The big problem that will be tackled is to create weather radar images over the ocean using an artificial neural network. First we will need to create them over the land with the radar, satellite and model weather data we have, to make sure we have a good working model with no errors that we can proceed to use for the radar images over the ocean with some extra modifications.

3. Background & Related Works

In order to tackle the problem previous research on different types of weather forecasting models was done by reading many different papers. In the first paper *Radar Measurement of Rainfall*, it is described how radar takes measurement of rainfall over time and some of the factors that can contribute to errors. Some of the errors can be caused by the measurement of radar reflectivity factor, evaporation and advection of precipitation before reaching the ground, and variations in the drop-size distribution and vertical air motions. In addition to talking about how errors can be caused by radar the literature explains some methods that can decrease the errors of the radar like having gage adjustments.

The paper *Rainfall retrieval over the ocean with spaceborne W-band radar* talks about a method for retrieving precipitation over the ocean using spaceborne W-band radar and applying it to the CloudSat Cloud Profiling Radar. The method is most applicable to stratiform-type precipitation. Measurements of radar backscatter from the ocean surface are combined with information about surface wind speed and sea surface temperature to derive the path-integrated attenuation through precipitating cloud systems. This paper gives an insight of using satellite images to predict rainfall.

The paper *Machine Learning for the Detection of Oil Spills in Satellite Radar Image*, uses an application of machine learning is used for the detection of oil spills from radar images of the sea surface. Although this article is not truly about weather, it does talk about the use of machine learning on radar images, which gives us an insight on something very similar to what we are going to do.

Another paper that uses neural networks for rainfall prediction is the paper *Rainfall Prediction using Artificial Neural Network on Map-Reduce Framework*. The paper talks about how with big data it becomes hard to use deterministic models since they are time consuming. It becomes hard to use them with large volumes of data, which is why machine learning is a good replacement. The paper proposes an approach of processing big volume of data using Hadoop. It will use Artificial Neural Network implemented with a Map-reduce framework, which will create short-term weather predications that are a day ahead. This is also something similar to our project that gives good insight on a different implementation.

The paper *A Deep Hybrid Model for Weather Forecasting*, talks about using a hybrid model that combines discriminatively trained predictive models with a deep neural network. The deep neural network models the joint statistics of a set of weather-related variables. The hybrid model was able to create better results than the NOAA benchmarks and for the future they want to see if they can increase it to more future predictions. This literature not only gives an insight of weather casting but it also gives an insight on benchmarks and how we can approach using them within our project.

The paper *Modeling and Prediction of Rainfall Using Radar Reflectivity Data: A Data-Mining Approach*, talks about a data mining approach to predict rainfall based on radar reflectivity and tipping-bucket data. They used neural network, random forest, classification and regression tree, support vector machine and k-nearest neighbor in order to build prediction models. After testing them the highest accuracy one was chosen for further study. The model that predicts rainfall from radar and TB data collected at Oxford performed better for future predictions.

The paper *Machine Learning Techniques for Short-Term Rain Forecasting System in the Northeastern Part of Thailand*, talks about using Decision Tree, Artificial Neural Network and Support Vector Machine to develop classification and prediction models for rainfall forecasts. In the literature it was shown how feature selection could be used to identify the relationships between rainfall occurrences and what models can be created and used for predicting accurate rainfall. The literature gave insight of the different machine learning used and comparing them with RMSE to see which one has a better result.

All of the papers that were read gave useful insights of not only different methods in order to go about predicting rainfall but also gave a good insight of ways to evaluate the work that is going to be done.

4. Methods

Before starting any work the group decided to create three different groups in which we would work together to come up with the best possible rainfall prediction algorithm. The three teams were the data management team, the algorithms team and the model evaluation team.

The data management team would process the radar, satellite and model data and would correlate them with each other based on time and area. The domain must be 31N-41N and 82W-102W with the time frequency being every hour of 2017. After the data was preprocessed and matched together it would be sent to the algorithms team that focused on implementing two different models pix2pix and vid2vid. They would get results that would be passed down and evaluated by the evaluation team that would also create simple models to use as a baseline.

Being a part of the evaluation team the focus was on creating a baseline model and having different methods of evaluating the different models. For the baseline it was decided to create a linear regression model that would have input as the pix2pix model 1, which had two satellite bands and four features from the model data. The specific inputs were satellite band 2, satellite band 6, visibility, relative Humidity, specific humidity and temperature. This was chosen because in order to compare the models we wanted to have the inputs the same to have fewer errors.

Since the linear regression model was going to have inputs similar to pix2pix the data the processed data that was give to the algorithms team from the data management team was also used for the linear regression. The processed data were PNG images of size 256x256 organized in two folders, A and B. A contains the inputs and B contains the radar images corresponding to the inputs. In the first attempt there were only 195 correlating images.

The first step was figuring out how to input the data. The chosen format was to stack the images and then feed it to the linear regression model one pixel at a time. The size of each image was 256*256 meaning that each image had 65,536 pixels. Multiplied by six since six images were getting trained into one radar image that meant 393,216 pixels were inputted to be trained to each radar image.

In doing so this gave more data to work with since we had a limited amount of correlating data at the time. Since all data was inputted a pixel at a time, the output would be a long pixel array so in order to recreate the outputs I had to go through every 65,536 pixels and reshape them to a 256*256 image.

In addition to getting some results first we looked into the feature coefficients in the linear regression model in order to find out which input was being more significant in creating a result.

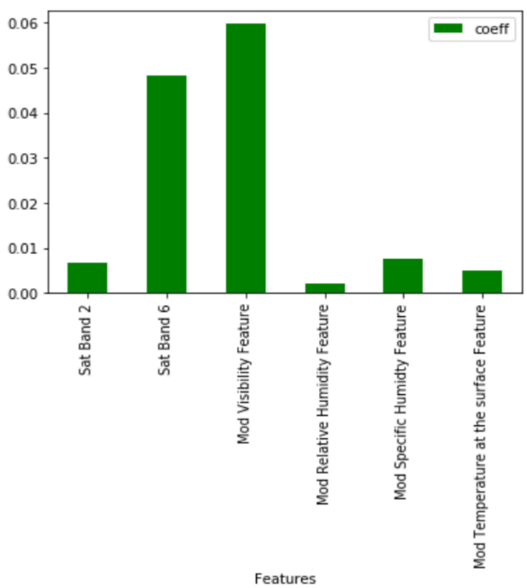

Fig.1 The chart above shows the coefficient for each input of the linear regression

After getting a small amount of outputs a evaluation was needed to be made in order to get a quick baseline. For the evaluation at first MSE and SSIM were chosen since they compared the similarity between two images. The mean squared error measures the average of the squares of the errors that is, the average squared difference between the estimated values and what is estimated.

For MSE and function was created that would intake two images and then compute the sum of the squared difference of the two images. For SSIM the SSIM function was used from the package Scikit-image that has many image processing algorithms.

At first the MSE and SSIM were giving okay results in comparing the images but more was need in order to actually evaluate the results. It was decided to create confusion matrices for the outputs of the different models since it would give us a more in depth idea if the pixel values were correct for each result.

In order to do the confusion matrix first the images needed to be changed to having one pixel value for clouds and one for none since I wanted to compare that clouds were actually predicted in the right place. Thresholding the image was the way to do this. In order to choose the threshold, I looked into the radar images distribution of pixel values.

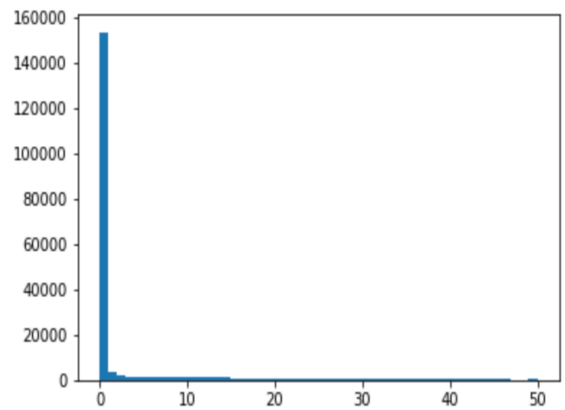

Fig 2. The graph above shows the distribution of pixel values at a radar image

Looking at the distribution figure it was seen that the change of pixel values occurred at the value one. So it was tested out and thresholding was done at one and also values from 0.5-15. At value one was were the threshold image would keep all the cloud information in the image while at values above greater than one the image would start to lose cloud information. So any pixel number below one would become zero while any pixel value equal to or greater than one would become 255. Zero being no cloud and 255 being cloud.

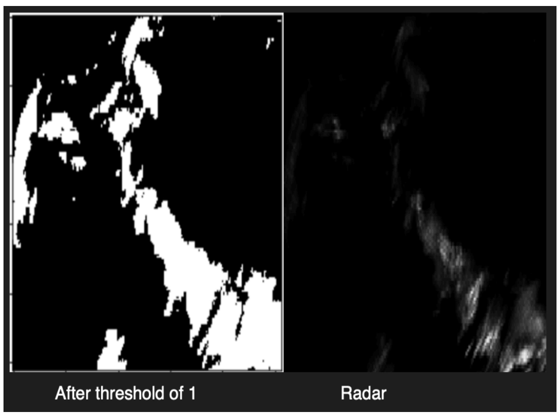

Fig.3 the threshold radar image vs. the original radar image

The image above shows one example of the thresholding being done to an image in it we can see that all of the cloud information is kept since even the faint grayish colors that are blending into the black are now fully shown in white.

Now that the threshold value was established the confusion matrix could be created. In order to do this the predicted images were flattened into an array of one dimension in order and the radar images were also flattened in another array in corresponding order. After having the two arrays of flattened images the confusion matrix algorithm was used from the package scikit-learn metrics. The algorithm would go through the two arrays and compare each pixel and organize it into true negative, false negative, true positive and false positive.

True negative is when it predicted no cloud correctly, false negative is when it predicted no cloud incorrectly, true positive is when it predicted a cloud correctly and false positive is when a cloud was predicted incorrectly. In order to get the percentage of each column of the confusion matrix the total number of pixels were counted and then divided each TN, FN, TP, and FP by it and multiplied by 100.

Other ways were used to also represent the confusion matrix like looking at a normalized confusion matrix but will be only showing the percentage one in this paper to give a clearer comparison and conclusion at the end.

After evaluation and comparing the linear regression to the first pix2pix model, I decided to play with the threshold of the linear regression model results because even though the results were way off at first glance the correct cloud was in it covered by lesser gray clouds. This created a significant discovery for linear regression.

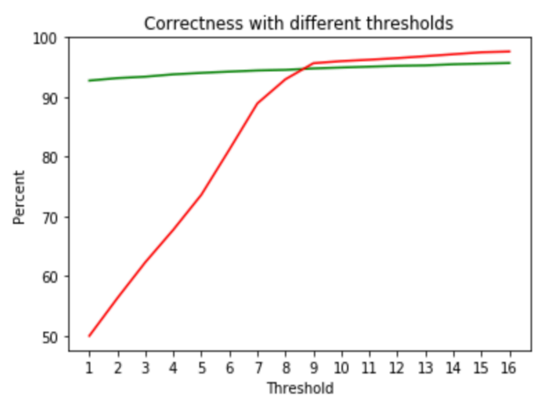

Fig.4 The figure above shows the linear regression correctness (red) vs. pix2pix model (green) at different thresholds

Changing the threshold of the linear regression to 15 gave a more accurate result from the confusion matrix since at that threshold the lesser significant clouds were removed and mostly left the cloud that was actually on the radar image. This gave an insight that linear regression could be able to predict rainfall but more results needed to be tested out since there were a few when this was discovered.

In addition I did more evaluations since the pix2pix algorithm team was trying many different models like only inputting satellite bands together with no model data, also inputting satellite bands individually. Another model that had augmented data as the inputs was also evaluated. In addition towards the end vid2vid algorithm team got some outputs, which were evaluated.

After evaluating all the different models and comparing the confusion matrix of each model the ones that will be looked at in the results and evaluation are the linear regression with and without thresholding, pix2px model 1 that had the same input as linear regression and model 2 which has only satellite bands as inputs and the vid2vid model.

5. Results & Evaluation

The first starting evaluation that was done was comparing the new model 1 pix2pix with an older model using MSE and SSIM since the baseline had yet to be created.

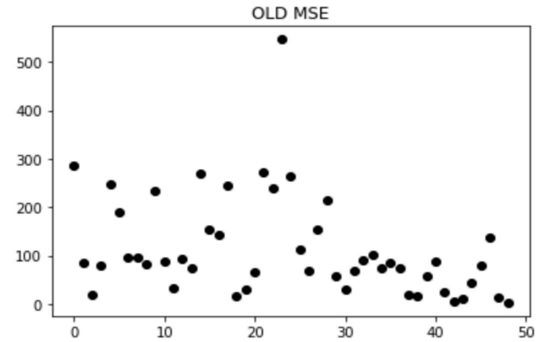

Fig. 5 MSE for older pix2pix from Capstone1

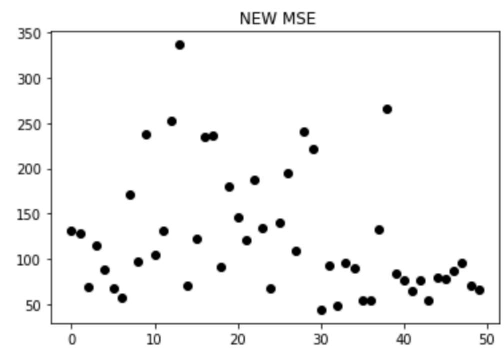

Fig.6 MSE for new pix2pix model 1

The above two figures show the MSE for two pix2pix models in which the x-axis is a pair of images predicted and corresponding radar and the y-axis is the MSE score they received. For MSE the closer the number is to zero the better the two images are the same. So in the two figures it shows that the new model was doing better since the highest MSE was less than 350 while for the old one it went over 500. In addition the concentration of scores is under 250 for the new model while for the old it is under 300. This was useful in the beginning of Capstone II since no baseline was yet achieved and the model showed improvement.

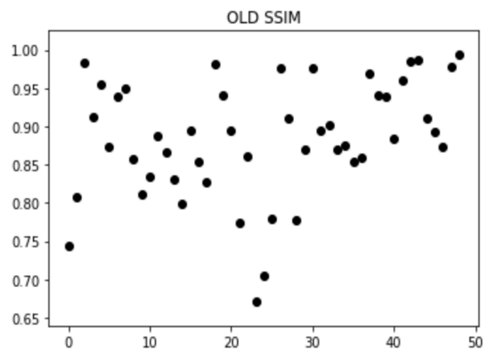

Fig.7 SSIM for older pix2pix from Capstone I

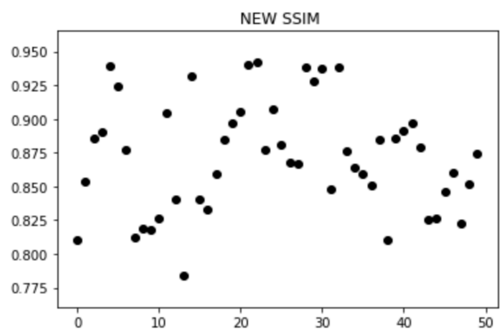

Fig.8 SSIM for new pix2pix model 1

The above two figures show the SSIM for two pix2pix models in which the x-axis is a pair of images predicted and corresponding radar and the y-axis is the SSIM score they received. For SSIM the closer to the number 1 the more similar the images are. The SSIM of the new one is not lower than 0.775 while the old one had 4 pairs below that and one pair being lower than 0.66. This shows an improvement in the new model.

The above evaluation were a small start into what really needed to be done like creating a simpler algorithm baseline to really compare instead of just comparing two pix2pix models. So after the linear regression model was created a new way of evaluating by using the confusion matrix was used with MSE and SSIM just on the side.

For the baseline a liner regression one pixel at a time with inputs satellite band 2, satellite band 6, visibility, relative Humidity, specific humidity and temperature was created. The first results were not that great since there were more clouds in it than there were in the radar image.

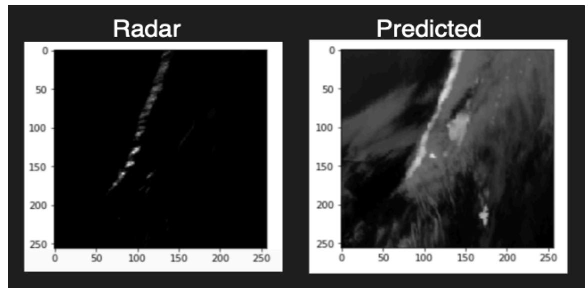

Fig.5 Radar image next to a predicted image by the Linear Regression

The first batch of results from the Linear Regression could be seen as bad at first glance since the predicted images had a lot of more clouds compared to the radar images as shown in the images above but we needed a numeric value as the baseline so the confusion matrix was calculated.

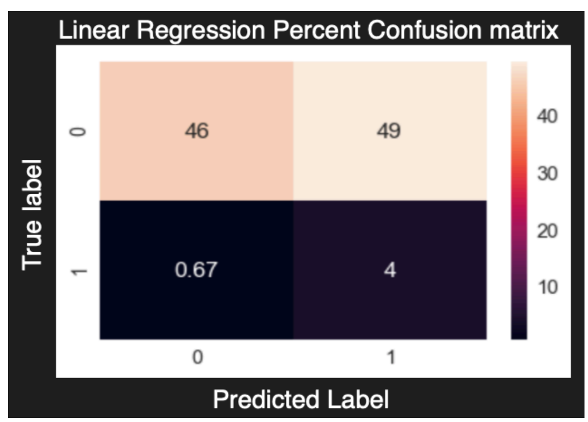

Fig.6 The confusion matrix of the Linear Regression

The figure above is the confusion matrix of the first baseline that was created. In it each pixel of the predicted image from linear regression model was compared to the real radar image. The numbers in the confusion matrix are the percentage of true negative, false negative, true positive and false positive for the images and sum up to 100. The overall baseline for this was a correctness of 50% and a wrongness of 49.67%, which is not a good score since it overall tells that the model is guessing the pixels since it has a 50% chance of choosing 0 or 1.

Now that a baseline was achieved, starting to evaluate the other models would be meaningful since a comparison could be made. The pix2pix pytorch model that had the same inputs as the linear regression baseline was then evaluated the same way and its confusion matrix is shown next.

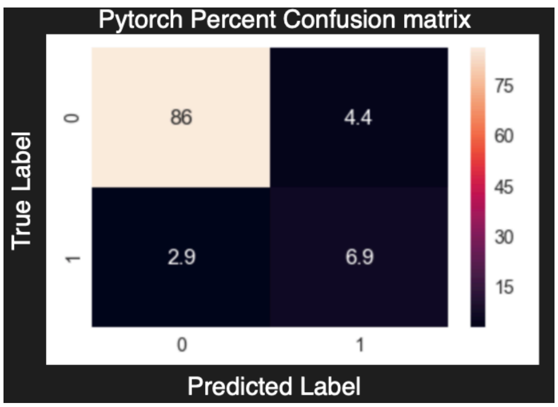

Fig.7 The confusion matrix of the Pix2Pix pytorch

The above figure shows that the Pix2Pix pytorch model had an overall correctness of 92.9% and a wrongness of 7.3% on the images that it outputted. This gave the team an insight that Pix2Pix was definitely performing better than other models since now the baseline had been created.

After realizing that changing the threshold affected the linear regression drastically new results were created.

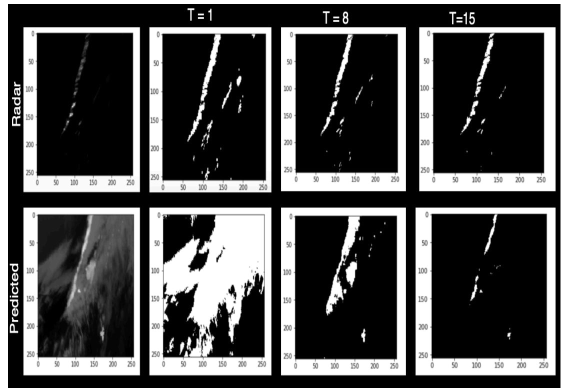

Fig.8 Linear regression images vs. radar at different thresholds

The above image shows how having a threshold of 15 instead of 1 removed the clouds that were predicted wrong by the model and kept only the ones that were correct. Noticing this drastic change a new confusion matrix was created for all the output images of the linear regression model but now with threshold 15.

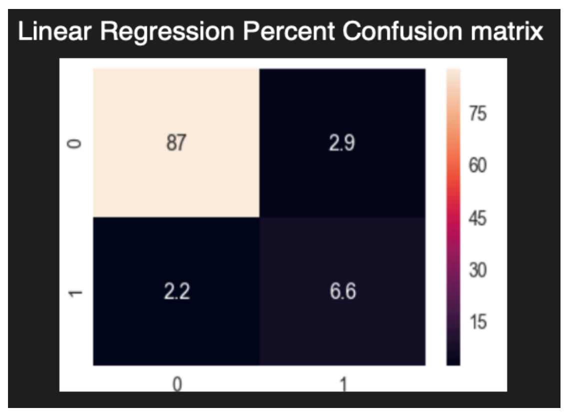

Fig.9 Confusion matrix of linear regression results with threshold 15

The new confusion matrix showed that thresholding the output images of the linear regression at 15 gave a correctness of 93.6% while a wrongness of 5.1%. This was very skeptical since linear regression was performing better than the pix2pix model with a correctness of 92.9%. On the other hand still had to keep in mind that overall pix2pix was still better since the predicted images were almost identical to the radar images at first glance while the linear regression model had to be altered through thresholding in order to get a good score of correctness. In addition the linear regression model had less outputs than the pix2pix.

As more outputs were predicted using the linear regression model the correctness started to decrease and got to a value of 89.6% and wrongness of 10.4%. This gave an insight that although the linear regression is getting good results overall the correctness had dropped under that of the pix2pix model making the pix2pix better.

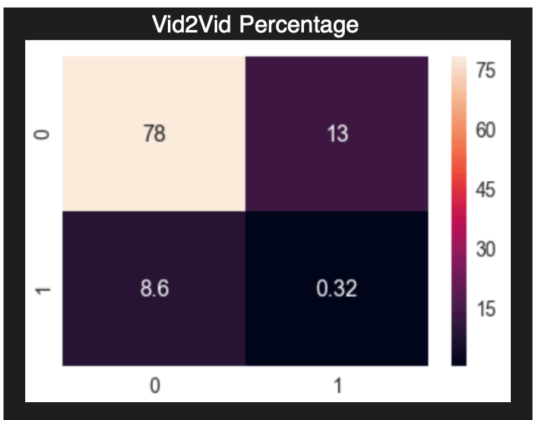

Fig. 10 Confusion matrix for Vid2Vid

The above figure shows the confusion matrix that was gotten from the vid2vid results. This gives the vid2vid a correctness of 78% with a wrongness of 22%. Even though the results seem good with a high correctness of 78% the model only had 30 predicted images and most of the pixels were 0 since the images were mostly black. The model was good for predicting no clouds but really bad at predicting clouds. Overall it performed badly compared to the other models that were created.

Model	Correctness	Wrongness
Linear regression	50%	49.67%
Linear regression T=15	89.6%	10.4%
Pix2Pix model 1	92.9%	7.3%
Pix2Pix model 2	89.1%	10.1%
Vid2Vid	78%	22%

Fig.10 Table showing the correctness of the different tested models

Even though many models were evaluated like trying pix2pix with single band inputs those models did really poorly and the graph above shows the percentage correctness for the 5 models that were more significant in my evaluation. The table above shows that the best model overall was the pix2pix model 1 that had the best correctness of 92.9%.

6. Conclusion

As a baseline the linear regression was a great start and in the end ended up performing extremely well. Although it did get up to 89.5% correctness, which is really well, the more outputs that were predicted it started to decrease it. More research needs to be done in order to take this significant finding as correct.

Furthermore when all models were evaluated the Pix2Pix model 1 did the best with an overall correctness of 92.89% and in the end is the model the group should continue to research and improve to solve the main problem of getting radar rainfall predictions over the ocean since it’s the best one at predicting rainfall clouds so far.

This was really interesting work done since it got not only a potential simple model the linear regression but also a more complex one the pix2pix model that can potentially predict rainfall since with the limited resources we were able to get great results.

7. References

Aditya Grover, Ashish Kapoor, and Eric Horvitz. 2015. A Deep Hybrid Model for Weather Forecasting. In Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '15). ACM, New York, NY, USA, 379-386. DOI: https://doi.org/10.1145/2783258.2783275

A. Kusiak, X. Wei, A. P. Verma and E. Roz, "Modeling and Prediction of Rainfall Using Radar Reflectivity Data: A Data-Mining Approach," in *IEEE Transactions on Geoscience and Remote Sensing*, vol. 51, no. 4, pp. 2337-2342, April 2013. doi: 10.1109/TGRS.2012.2210429

Haynes, J. M., T. S. L'Ecuyer, G. L. Stephens, S. D. Miller, C. Mitrescu, N. B. Wood, and S. Tanelli (2009), Rainfall retrieval over the ocean with spaceborne W-band radar, *J. Geophys. Res.*, 114, D00A22, doi:10.1029/2008JD009973.

Ingsrisawang, L. , Ingsriswang, S. , Somchit, S. , Aungsuratana, P. , Warawut Khantiyanan, W. (2008). 'Machine Learning Techniques for Short-Term Rain Forecasting System in the Northeastern Part of Thailand'. World Academy of Science, Engineering and Technology, International Science Index 17, International Journal of Computer, Electrical, Automation, Control and Information Engineering, 2(5), 1422 - 1427.

Kubat, M., Holte, R.C. & Matwin, S. Machine Learning (1998) 30: 195. https://doi.org/10.1023/A:1007452223027

K. Namitha, A. Jayapriya, and G. Santhosh Kumar. 2015. Rainfall Prediction using Artificial Neural Network on Map-Reduce Framework. In Proceedings of the Third International Symposium on Women in Computing and Informatics (WCI '15), Indu Nair (Ed.). ACM, New York, NY, USA, 492-495. DOI: http://dx.doi.org/10.1145/2791405.2791468

Sumi, S., Zaman, M., & Hirose, H. (2012). A rainfall forecasting method using machine learning models and its application to the Fukuoka city case, *International Journal of Applied Mathematics and Computer Science*, *22*(4), 841-854. doi: https://doi.org/10.2478/v10006-012-0062-1

Wilson, J.W. and E.A. Brandes, 1979: Radar Measurement of Rainfall—A Summary. *Bull. Amer. Meteor. Soc.,* **60**, 1048–1060, https://doi.org/10.1175/1520-0477(1979)060<1048:RMORS>2.0.CO;2