Using Machine Learning Techniques to Improve Precipitation Forecasting

Joshua Coblenz

***Abstract*—This paper studies the effect of machine learning techniques on precipitation forecasting. Twelve features from the global forecast system (GFS) numerical weather prediction model are used to classify a precipitation estimate. Data are chosen from six and twenty-four hour forecasts in two locations: near Washington DC and near Seattle. Machine learning algorithms are applied to binary classification, where the response variable is called precipitation or no precipitation, and multi-class classification, where the response variable is divided into five precipitation levels. Single techniques, as well as an ensembling technique called random forests, are investigated.**

I. Introduction and Related Work

Numerical weather prediction (NWP) has long been a difficult task for meteorologists. Atmospheric dynamics is extremely complicated to model, and chaos theory teaches us that the mathematical equations used to predict the weather are sensitive to initial conditions; that is, slightly perturbed initial conditions could yield very different forecasts.

Over the years, meteorologists have developed a number of different mathematical models for atmospheric dynamics, each making slightly different assumptions and simplifications, and hence each yielding different forecasts. It has been noted that each model has its strengths and weaknesses forecasting in different situations, and hence to improve performance, scientists now use an ensemble forecast consisting of different models and running those models with different initial conditions. This ensemble method uses statistical post-processing (usually linear regression) of the ensemble members [4].

Recently, machine learning techniques have started to be applied to NWP. Studies of neural networks, logistic regression, and genetic algorithms have shown improvements over standard linear regression for precipitation prediction [1]. Gagne et al proposed using multiple machine learning techniques to improve precipitation forecasting [3]. They used Breiman's random forest technique [2], which had previously been applied to other areas of meteorology, including aviation turbulence [5], to learn from the CAPS storm-scale ensemble forecast (SSEF) data. Performance was verified using next generation weather radar (NEXRAD) data.

Instead of using an ensemble forecast, this paper discusses the use of machine learning techniques to improve the precipitation forecast from one ensemble member.

II. Data

A. Input Variables

The global forecast system (GFS) is a numerical weather prediction system with a horizontal resolution of a half of a degree, which divides the atmosphere into 64 vertical layers [6]. See [7] for documentation on the model. The GFS model is run four times daily, with forecasts from 3 to 180 hours in increments of 3 hours. The national oceanic and atmospheric administration (NOAA) stores historical GFS forecasts in a database called the national operational model archive and distribution system (NOMADS) [8]. This paper applies machine learning to the 6-hour and 24-hour forecasts for all four forecasting cycles.

The GFS model forecasts many variables. In this project, 12 forecast variables were chosen as features in the machine learning algorithms. These were relative humidity, specific humidity, vertical velocity, precipitation rate, surface pressure, U wind component, V wind component, convective precipitation, precipitable water, cloud water, helicity, and total precipitation.

All forecast data from 2014 were collected at two locations: 39N 77W and 48.5N 122.5W. These locations correspond roughly to Washington DC and Seattle, which were chosen because precipitation forecasting at each location is challenging, yet the weather at the two stations is dissimilar.

B. Response Variable

In 1988, NOAA established the WSR-88D weather surveillance Radar Operations Center. The center derives precipitation estimates from radar reflectivity from the Next Generation Weather Radar (NEXRAD) network [9]. The response variable ("truth labels") in this project is the 'precipitation estimate' NEXRAD Level III product.

C. Data Conditioning

The 6-hour and 24-hour forecasts of total precipitation used as a feature consisted of predictions of total precipitation over 6 hours. That is, the 6-hour forecast was a prediction of total precipitation for hours 0-6, and the 24-hour forecast was a prediction of total precipitation for hours 18-24. The NEXRAD precipitation output, however, was the total precipitation for the previous 3 hours. Therefore, the outputs from 2 consecutive 3-hour precipitation data points were added together to form a 6-hour precipitation history. Obviously, the input and response data were only used if the forecast and both corresponding 3-hour precipitation estimates existed.

III. Experiment One: Binary Classification

The first part of this study concerned binary classification: precipitation or no precipitation. Data where there was no precipitation were labeled 0 while data where there was precipitation were labeled 1. Four different machine learning

TABLE I
Error for Each Fold

	KNN	NBC	SVM	TREE
1	0.47	0.29	0.43	0.29
2	0.36	0.34	0.46	0.32
3	0.43	0.30	0.47	0.28
4	0.36	0.35	0.32	0.27
5	0.36	0.33	0.47	0.29
6	0.45	0.30	0.45	0.31
7	0.43	0.35	0.43	0.34
8	0.37	0.30	0.42	0.31
9	0.45	0.33	0.35	0.27
10	0.39	0.35	0.35	0.21
11	0.45	0.31	0.48	0.35
12	0.46	0.33	0.44	0.40

TABLE II
KNN Confusion Matrix

	No Precip	Precip
Predict No Precip	472	289
Predict Precip	260	302

TABLE III
NBC Confusion Matrix

	No Precip	Precip
Predict No Precip	689	383
Predict Precip	43	208

TABLE IV
SVM Confusion Matrix

	No Precip	Precip
Predict No Precip	480	307
Predict Precip	252	284

TABLE V
TREE Confusion Matrix

	No Precip	Precip
Predict No Precip	545	214
Predict Precip	187	377

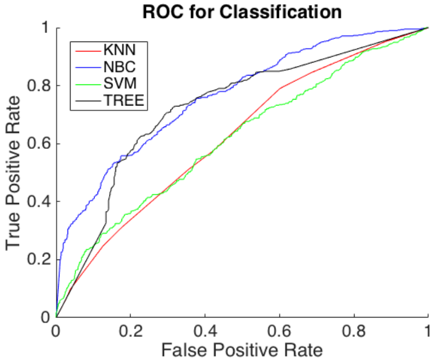

Fig. 1. ROC Curves for Different Learning Techniques

TABLE VI
2 Tree Confusion Matrix

	No Precip	Precip
Predict No Precip	718	124
Predict Precip	14	467

TABLE VII
5 Tree Confusion Matrix

	No Precip	Precip
Predict No Precip	722	35
Predict Precip	10	556

techniques were applied to the Washington DC 6-hour forecast: the 5-nearest-neighbor method (KNN), the naive Bayes classifier (NBC), the support vector machine method (SVM), and the classification tree method (TREE). Each method was cross-validated with 12 folds (corresponding roughly to the 12 months in a year), and the error was calculated for each fold. Table 1 shows the error for each method for each fold. Note that within a model, the error for each fold is similar.

Tables 2 through 5 show the confusion matrices for each method. For binary classification, the confusion matrix is a 2x2 matrix, where the top left entry is the number of data points where the model predicted no precipitation and there actually was no precipitation. The top right entry is the number of data points where the model predicted no precipitation but there actually was precipitation. The bottom left entry is the number of data points where the model predicted precipitation but there actually was no precipitation. The bottom right entry is the number of data points where the model predicted precipitation and there actually was precipitation. The perfect outcome would be where the diagonal entries are nonzero and the off-diagonal entries are zero.

Figure 1 shows the receiver operating characteristic (ROC) curves for the four machine learning methods. In the ROC curve, the x-axis is the false positive rate (i.e. predicting precipitation when it didn't precipitate), and the y-axis is the true positive rate (i.e. predicting precipitation when it did precipitate). The ROC curves show the tradeoff in a model between correctly predicting precipitation when it truly does precipitate and predicting precipitation when it truly does not precipitate. Since further to the top-left is better (i.e correctly predicting precipitation without falsely predicting precipitation), one can see that the naive Bayes classifier and tree method were the best methods. However, as seen from the confusion matrices, none of these four methods yielded particularly great results. In fact, machine learning methods composed of an ensemble of predictions should do better.

One ensemble method in machine learning called Random Forest was introduced by Leo Breiman in 2001 [2]. It consists of an ensemble of different decision trees. The idea is that the training data are bootstrap sampled with replacement for each tree, and voting among trees yields the final classification. An input to the random forest algorithm is the number of different trees in the ensemble. A random forest consisting of 2, 5, 10, and 20 trees was applied to the Washington DC 6-hour forecast. Tables 6 through 9 show the confusion matrices for each number of trees.

Figure 2 shows the ROC curve for the different number

TABLE VIII
10 Tree Confusion Matrix

	No Precip	Precip
Predict No Precip	728	21
Predict Precip	4	570

TABLE IX
20 Tree Confusion Matrix

	No Precip	Precip
Predict No Precip	732	6
Predict Precip	0	585

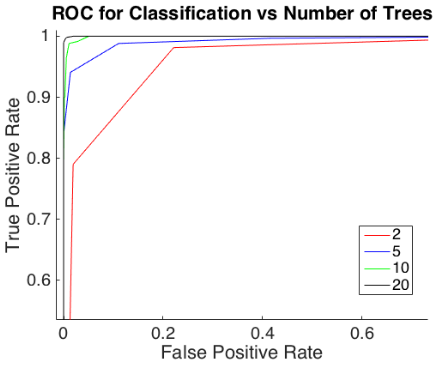

Fig. 2. ROC Curves for Different Numbers of Trees in Ensemble

TABLE X
6 hr Washington DC Confusion Matrix

	No Precip	Precip
Predict No Precip	729	13
Predict Precip	3	578

TABLE XI
6 hr Seattle Confusion Matrix

	No Precip	Precip
Predict No Precip	714	8
Predict Precip	2	550

TABLE XII
24 hr Washington DC Confusion Matrix

	No Precip	Precip
Predict No Precip	720	18
Predict Precip	7	575

TABLE XIII
24 hr Seattle Confusion Matrix

	No Precip	Precip
Predict No Precip	708	12
Predict Precip	5	546

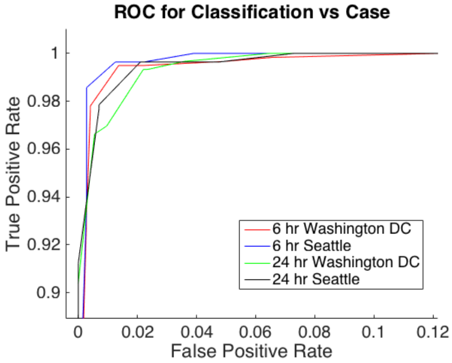

Fig. 3. ROC Curves for Different Cases

TABLE XIV
6 hr Seattle trained on 6 hr Seattle Confusion Matrix

	No Precip	Precip
Predict No Precip	712	23
Predict Precip	4	535

TABLE XV
6 hr Seattle trained on 6 hr DC Confusion Matrix

	No Precip	Precip
Predict No Precip	387	133
Predict Precip	329	425

TABLE XVI
24 hr DC trained on 24 hr DC Confusion Matrix

	No Precip	Precip
Predict No Precip	722	21
Predict Precip	5	572

of trees. As expected, the ensembling method improves performance, and the larger the number of trees, the better the performance.

All the above results were obtained by applying machine learning algorithms to the 6-hour forecast near Washington DC. However, all four cases (6-hour Washington DC, 6-hour Seattle, 24-hour Washington DC, and 24-hour Seattle) can be compared. Tables 10 through 13 show the confusion matrix for each case, where the random forest technique consisting of 10 trees was applied to each.

Figure 3 shows the ROC curve for the different cases. As expected, the performance of the learned 6-hour forecasts was better than the performance of the learned 24-hour forecasts. It was also interesting to note that the Seattle performance was better than the Washington DC performance for both the 6-hour and 24-hour forecasts.

How well the models generalize to data from the other cases was studied as well. Tables 14 through 17 show a comparison of the confusion matrices where the random forest was trained and tested with the same data, to where the random forest was trained with data from one case and tested with data from another case. To generate Table 15, the random forest was trained with the 6-hour Washington DC data and tested with the 6-hour Seattle data. Table 17 was generated by training with the 6-hour Washington DC data and testing with the 24-hour Washington DC data.

Figure 4 shows the ROC curves when the model was trained on one data set and tested on another. As expected, training on the 6-hour Washington DC data and testing on the 6-hour Seattle data was worse than training and testing on the 6-

TABLE XVII
24 hr DC trained on 6 hr DC Confusion Matrix

	No Precip	Precip
Predict No Precip	482	281
Predict Precip	245	312

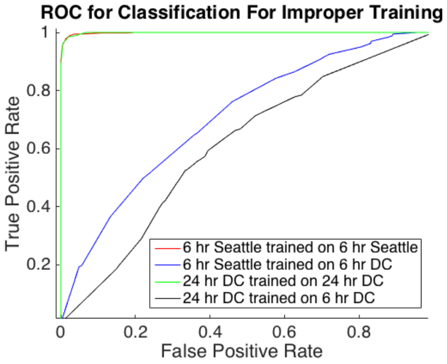

Fig. 4. ROC Curves for Different Cases

TABLE XVIII
KNN Confusion Matrix

	Cat 1	Cat 2	Cat 3	Cat 4	Cat 5
Predict Cat 1	540	152	196	8	1
Predict Cat 2	82	15	42	5	0
Predict Cat 3	110	46	110	7	5
Predict Cat 4	0	0	2	0	1
Predict Cat 5	0	0	1	0	0

TABLE XIX
NBC Confusion Matrix

	Cat 1	Cat 2	Cat 3	Cat 4	Cat 5
Predict Cat 1	683	159	215	2	0
Predict Cat 2	33	124	38	3	0
Predict Cat 3	14	20	75	7	0
Predict Cat 4	2	9	17	6	5
Predict Cat 5	0	1	6	2	2

hour Seattle data. Also as expected, training on the 6-hour Washington DC data and testing on the 24-hour Washington DC data was worse than training and testing on the 24 hour Washington DC data. What was surprising was that changing forecast hours between training and testing was worse than changing stations between training and testing.

IV. Experiment Two: Multi-class Classification

The second part of this study concerned multi-class classification. The precipitation values were split into five categories: 0, 0 to 0.1, 0.1 to 0.5, 0.5 to 1, and greater than 1. Again, non-ensembling machine learning methods were first used. The 5 nearest neighbor, naive Bayes, and decision tree methods were applied. Tables 18 through 20 show the confusion matrices for each method.

TABLE XX
TREE Confusion Matrix

	Cat 1	Cat 2	Cat 3	Cat 4	Cat 5
Predict Cat 1	501	93	104	6	1
Predict Cat 2	119	56	83	2	1
Predict Cat 3	110	59	156	8	1
Predict Cat 4	2	5	7	3	4
Predict Cat 5	0	0	1	1	0

TABLE XXI
2 Tree Confusion Matrix

	Cat 1	Cat 2	Cat 3	Cat 4	Cat 5
Predict Cat 1	725	60	78	4	0
Predict Cat 2	5	150	48	3	2
Predict Cat 3	2	3	225	7	3
Predict Cat 4	0	0	0	6	1
Predict Cat 5	0	0	0	0	1

TABLE XXII
5 Tree Confusion Matrix

	Cat 1	Cat 2	Cat 3	Cat 4	Cat 5
Predict Cat 1	726	25	21	0	0
Predict Cat 2	2	184	7	0	0
Predict Cat 3	4	4	322	1	1
Predict Cat 4	0	0	1	19	1
Predict Cat 5	0	0	0	0	5

TABLE XXIII
10 Tree Confusion Matrix

	Cat 1	Cat 2	Cat 3	Cat 4	Cat 5
Predict Cat 1	732	17	6	0	0
Predict Cat 2	0	191	1	0	0
Predict Cat 3	0	5	343	0	0
Predict Cat 4	0	0	0	20	0
Predict Cat 5	0	0	1	0	7

TABLE XXIV
20 Tree Confusion Matrix

	Cat 1	Cat 2	Cat 3	Cat 4	Cat 5
Predict Cat 1	732	1	2	0	0
Predict Cat 2	0	212	0	0	0
Predict Cat 3	0	0	349	2	0
Predict Cat 4	0	0	0	18	0
Predict Cat 5	0	0	0	0	7

The percent correct classification (Pcc) for each model was calculated as the sum of the diagonals of the confusion matrix divided by the total number of data points. Pcc for KNN was 0.57; Pcc for NBC was 0.68; and Pcc for TREE was 0.54. Therefore one can conclude that the naive Bayes classifier worked the best. It is also interesting to note that naive Bayes is the only classifier that correctly classified any Category 5 (1+) data points.

Ensembling in the form of the random forest technique was also applied to the multi-class data. The study of the effect of the number of trees in the ensemble was again performed. As before, random forests with 2, 5, 10, and 20 trees were applied. Tables 21 through 24 show the confusion matrices for each number of trees. The Pcc numbers for the four cases were 0.84, 0.95, 0.98, and 0.996, respectively.

The random forest technique was again applied to all four cases (6-hour Washington DC, 6-hour Seattle, 24-hour Washington DC, and 24-hour Seattle). Tables 25 through 28 show the confusion matrices for each case. The Pcc for each

TABLE XXV
6 HR WASHINGTON DC CONFUSION MATRIX

	Cat 1	Cat 2	Cat 3	Cat 4	Cat 5
Predict Cat 1	731	8	7	1	0
Predict Cat 2	1	201	2	0	0
Predict Cat 3	0	4	342	0	1
Predict Cat 4	0	0	0	19	0
Predict Cat 5	0	0	0	0	6

TABLE XXVI
6 HR SEATTLE CONFUSION MATRIX

	Cat 1	Cat 2	Cat 3	Cat 4	Cat 5
Predict Cat 1	713	14	5	0	0
Predict Cat 2	1	214	0	0	0
Predict Cat 3	2	1	322	1	0
Predict Cat 4	0	0	0	1	0
Predict Cat 5	0	0	0	0	0

TABLE XXVII
24 HR WASHINGTON DC CONFUSION MATRIX

	Cat 1	Cat 2	Cat 3	Cat 4	Cat 5
Predict Cat 1	724	13	9	1	0
Predict Cat 2	2	197	0	0	0
Predict Cat 3	1	2	345	0	1
Predict Cat 4	0	0	0	19	0
Predict Cat 5	0	0	0	0	6

TABLE XXVIII
24 HR SEATTLE CONFUSION MATRIX

	Cat 1	Cat 2	Cat 3	Cat 4	Cat 5
Predict Cat 1	713	11	7	0	0
Predict Cat 2	0	216	0	0	1
Predict Cat 3	0	1	320	0	0
Predict Cat 4	0	0	0	2	0
Predict Cat 5	0	0	0	0	0

TABLE XXIX
6 HR SEATTLE TRAINED ON 6 HR SEATTLE CONFUSION MATRIX

	Cat 1	Cat 2	Cat 3	Cat 4	Cat 5
Predict Cat 1	712	8	8	0	0
Predict Cat 2	1	218	0	0	0
Predict Cat 3	3	3	319	0	0
Predict Cat 4	0	0	0	2	0
Predict Cat 5	0	0	0	0	0

case was 0.98.

Model generalization to data from the other cases was also studied for the multi-class experiment. Tables 29 through 32 show the confusion matrices for the comparison between the random forest being trained and tested on the same data, and being trained with data from one case and tested with data from another case. Again, to generate Table 30, the random forest was trained with the 6-hour Washington DC data and tested with the 6-hour Seattle data. Table 32 was generated by training with the 6-hour Washington DC data and testing with the 24-hour Washington DC data. The Pcc values for the cases were 0.98, 0.99, 0.55, and 0.49, respectively. As with the binary classification experiment, training on 6-hour and testing on 24-hour Washington DC data was worse than training on 6-hour Washington DC data and testing on 6-hour Seattle data.

TABLE XXX
6 HR SEATTLE TRAINED ON 6 HR WASHINGTON DC CONFUSION MATRIX

	Cat 1	Cat 2	Cat 3	Cat 4	Cat 5
Predict Cat 1	490	123	90	0	0
Predict Cat 2	54	24	31	0	0
Predict Cat 3	172	82	200	2	0
Predict Cat 4	0	0	6	0	0
Predict Cat 5	0	0	0	0	0

TABLE XXXI
24 HR WASHINGTON DC TRAINED ON 24 HR WASHINGTON DC CONFUSION MATRIX

	Cat 1	Cat 2	Cat 3	Cat 4	Cat 5
Predict Cat 1	727	8	5	0	0
Predict Cat 2	0	204	0	0	0
Predict Cat 3	0	0	349	0	0
Predict Cat 4	0	0	0	20	0
Predict Cat 5	0	0	0	0	7

TABLE XXXII
24 HR WASHINGTON DC TRAINED ON 6 HR WASHINGTON DC CONFUSION MATRIX

	Cat 1	Cat 2	Cat 3	Cat 4	Cat 5
Predict Cat 1	477	109	153	8	4
Predict Cat 2	112	31	55	6	1
Predict Cat 3	125	66	142	5	2
Predict Cat 4	9	5	4	1	0
Predict Cat 5	4	1	0	0	0

TABLE XXXIII
VARIABLE IMPORTANCE

	6 hr DC	6 hr Seattle	24 hr DC	24 hr Seattle
Var 1	0.76	1.27	-0.20	0.33
Var 2	0.79	0.54	1.07	1.26
Var 3	0.85	0.76	0.21	1.30
Var 4	2.20	0.73	-0.05	1.13
Var 5	1.66	1.34	1.39	1.57
Var 6	1.00	1.30	0.21	0.80
Var 7	2.00	1.00	2.21	2.67
Var 8	0.92	1.32	1.09	1.11
Var 9	2.32	1.44	0.59	1.02
Var 10	1.82	0.73	0.31	0.63
Var 11	1.00	0.50	0.95	0.66
Var 12	1.04	1.29	0.74	1.54

V. EXPERIMENT 3: DECREASING NUMBER OF FEATURES

One of the features of the MATLAB implementation of the random forest technique is that it calculates the relative importance of each variable. This is done by measuring the increase in prediction error if the values of that variable are permuted across the out-of-bag observations (those not used in training each tree). This measure is computed for every tree, then averaged over the entire ensemble and divided by the standard deviation over the entire ensemble. Table 33 shows the variable importance for the four cases (6-hour and 24-hour Washington DC, and 6-hour and 24-hour Seattle).

One might imagine that the dimensionality of the problem could be reduced by only utilizing the few most important variables. However, running the random forest algorithm again, such that the forest consisted of different trees trained from different subsets of the data, yielded variable importance

TABLE XXXIV
Variable Importance, Round 2

	6 hr DC	6 hr Seattle	24 hr DC	24 hr Seattle
Var 1	0.85	0.25	0.28	-0.14
Var 2	0.43	0.97	0.70	0.87
Var 3	1.12	1.47	2.31	0.72
Var 4	2.48	1.42	0.98	0.47
Var 5	1.81	1.13	1.97	0.96
Var 6	2.32	1.25	0.55	0.70
Var 7	1.98	1.28	2.43	1.07
Var 8	0.46	1.66	1.49	0.26
Var 9	1.47	0.66	1.20	0.12
Var 10	0.58	-0.06	0.58	0.89
Var 11	0.84	1.09	1.21	0.33
Var 12	0.45	1.67	0.53	0.88

TABLE XXXV
6 hr Washington DC Confusion Matrix

	Cat 1	Cat 2	Cat 3	Cat 4	Cat 5
Predict Cat 1	730	9	7	0	0
Predict Cat 2	0	199	1	0	0
Predict Cat 3	2	4	343	0	1
Predict Cat 4	0	1	0	20	0
Predict Cat 5	0	0	0	0	6

TABLE XXXVI
6 hr Seattle Confusion Matrix

	Cat 1	Cat 2	Cat 3	Cat 4	Cat 5
Predict Cat 1	713	11	4	0	0
Predict Cat 2	0	218	0	0	0
Predict Cat 3	3	0	323	0	0
Predict Cat 4	0	0	0	2	0
Predict Cat 5	0	0	0	0	0

TABLE XXXVII
24 hr Washington DC Confusion Matrix

	Cat 1	Cat 2	Cat 3	Cat 4	Cat 5
Predict Cat 1	725	6	16	1	1
Predict Cat 2	0	205	1	0	0
Predict Cat 3	2	1	337	2	0
Predict Cat 4	0	0	0	17	0
Predict Cat 5	0	0	0	0	6

TABLE XXXVIII
24 hr Seattle Confusion Matrix

	Cat 1	Cat 2	Cat 3	Cat 4	Cat 5
Predict Cat 1	708	12	7	0	0
Predict Cat 2	0	216	2	1	0
Predict Cat 3	5	0	318	0	0
Predict Cat 4	0	0	0	1	0
Predict Cat 5	0	0	0	0	1

shown in Table 34. Note that the top 4 variables in each run aren't the same, so there was clear choice of which variables to ignore.

Nevertheless, in the name in science, the impact of reducing variables was studied. Variables 1, 2, 8, 9, and 12 were removed, and confusion matrices were calculated for each case (Tables 35 through 38). Again, the Pcc numbers turned out to all be 0.98, so removing those 5 variables did not have much of an effect on the training error.

TABLE XXXIX
6 hr Seattle trained on 6 hr Seattle Confusion Matrix

	Cat 1	Cat 2	Cat 3	Cat 4	Cat 5
Predict Cat 1	711	8	4	0	0
Predict Cat 2	2	218	0	0	0
Predict Cat 3	3	3	323	0	0
Predict Cat 4	0	0	0	2	0
Predict Cat 5	0	0	0	0	0

TABLE XL
6 hr Seattle trained on 6 hr Washington DC Confusion Matrix

	Cat 1	Cat 2	Cat 3	Cat 4	Cat 5
Predict Cat 1	388	89	58	0	0
Predict Cat 2	96	34	44	0	0
Predict Cat 3	229	103	193	1	0
Predict Cat 4	3	3	32	1	0
Predict Cat 5	0	0	0	0	0

TABLE XLI
24 hr Washington DC trained on 24 hr Washington DC Confusion Matrix

	Cat 1	Cat 2	Cat 3	Cat 4	Cat 5
Predict Cat 1	724	15	10	0	1
Predict Cat 2	0	196	0	0	0
Predict Cat 3	2	1	344	0	0
Predict Cat 4	1	0	0	20	0
Predict Cat 5	0	0	0	0	6

TABLE XLII
24 hr Washington DC trained on 6 hr Washington DC Confusion Matrix

	Cat 1	Cat 2	Cat 3	Cat 4	Cat 5
Predict Cat 1	474	117	157	8	3
Predict Cat 2	110	27	56	6	2
Predict Cat 3	130	62	136	5	2
Predict Cat 4	9	5	4	1	0
Predict Cat 5	4	1	1	0	0

Model application to data from the other cases was also studied again. Tables 36 through 39 show the confusion matrices for the comparison between the random forest being trained and tested on the same data, and the random forest being trained with data from one case and tested with data from another case. Again, to generate Table 37, the random forest was trained with the 6-hour Washington DC data and tested with the 6-hour Seattle data. Table 39 was generated by training with the 6-hour Washington DC data and testing with the 24-hour Washington DC data. The Pcc values for the cases were 0.98, 0.98, 0.48, and 0.48, respectively. Yet again, training on 6-hour and testing on 24-hour Washington DC data was worse than training on 6-hour Washington DC data and testing on 6-hour Seattle data.

VI. Summary and Future Work

This project has studied the application of machine learning techniques on precipitation forecasting using variables from the GFS numerical weather prediction model as features. First, a number of binary classification algorithms were applied. These included the k nearest neighbors algorithm, a naive Bayes classifier, the support vector technique, and a decision tree. Each of these models was 12-fold cross validated, and it was shown that the performance on each of the folds was similar. For the data sets used, the naive Bayes classifier and the single decision tree performed the best. This was evidenced by the confusion matrix and ROC curve for each technique (Tables 2 through 5 and Figure 1).

In order to improve performance, an ensemble of machine learning techniques was applied. This was in the form of a random forest, consisting of multiple trees trained from subsets of the data, chosen with replacement. Performance of the random forest versus number of trees in the forest was calculated, and as expected, training error decreased as the number of trees increased (Tables 6 through 9 and Figure 2).

Next, a random forest consisting of 10 trees was applied to four data sets: near Washington DC from a 6-hour forecast, near Washington DC from a 24-hour forecast, near Seattle from a 6-hour forecast, and near Seattle from a 24-hour forecast. Not surprisingly, the performance of machine learning on both 24-hour forecasts was worse than the performance on both 6-hour forecasts. It also turned out that performance on the Washington DC data was worse than the performance on Seattle data (Tables 10 through 13 and Figure 3).

The next step was to study how well the models generalized. This was done by four tests: training and testing on Seattle 6-hour data, training and testing on Washington DC 24-hour data, training on Washington DC 6-hour data and testing on Seattle 6-hour data, and training on Washington DC 6-hour data and testing on Washington DC 24-hour data. Also not surprisingly, training and testing on the same data well outperformed training on one set and testing on another. What was surprising was that training and testing on the same location but a different forecasting time performed worse than training and testing on two different locations but the same forecasting time (Tables 14 through 17 and Figure 4).

The next experiment studied machine learning techniques applied to a multi-class classification problem. Precipitation amount was arbitrarily split into 5 bins. The k nearest neighbors algorithm, a naive Bayes classifier, and a decision tree were applied. All techniques were 12-fold cross validated. By calculating the probability of correct classification (the sum of the diagonal entries in the confusion matrix divided by the total number of data points), it was shown that the naive Bayes classifier performed the best (Tables 18 through 20). Ensembling using random forests was then applied, and the performance versus number of trees was investigated. As expected and as in the binary classification experiment, performance increased as the number of trees increased (Tables 21 through 24).

The random forest technique with 10 trees for multi-class classification was applied to all four data sets. Interestingly, all four data sets seemed to perform roughly the same (Tables 25 through 28). Next, as with binary classification, the effect of training on one data set and testing on another was studied. The same four tests were performed (training and testing on Seattle 6-hour data, training and testing on Washington DC 24-hour data, and training on Washington DC 6-hour data and testing on both Seattle 6-hour and Washington DC 24-hour data). The results were consistent with binary classification: training and testing on different data performed worse than training and testing on the same data, and switching forecasting time was worse than switching location (Tables 29 through 32).

Lastly, feature importance was studied. MATLAB's implementation of the random forest calculates the relative importance of the features. It was hoped that this would shed light on which of the 12 features were most important, such that the dimensionality of the problem could be reduced. However, running the random forest algorithm multiple times, yielding different trees trained from different subsets of the data, led to differing measures of feature importance (Tables 33 and 34). Nevertheless, 5 features were removed at random, and the machine learning was applied again. The removal of the features did not seem to significantly negatively affect training error for the 4 cases (Tables 35 through 38). However, the performance when training on one data set and testing on another did seem to go down slightly (Tables 39 through 42).

In the future, it would be interesting to delve more deeply into which features were most important, as well as why there seemed to be so much variation in the feature importance. The performance as a function of time of year could be studied. Machine learning algorithms could be applied to other numerical weather prediction models, such as the NAM. Also, the reason for such an effect of training on 6 hour forecasts and testing on 24 hour forecasts could be investigated.

References

[1] S. Applequist, G. Gahrs, R. Peffer, and X Niu; *Comparison of Methodologies for Probabilistic Quantitative Precipitation Forecasting*. Weather and Forecasting, Volume 17, pages 783-799, 2002.

[2] L. Breiman; *Random Forests*. Machine Learning, Volume 45, pages 5-32, 2001.

[3] D. Gagne, A. McGovern, and M. Xue; *Machine Learning Enhancement of Storm-Scale Ensemble Probabilistic Quantitative Precipitation Forecasts*. Weather and Forecasting, Volume 29, pages1024-1043, 2014.

[4] M. Tracton, and E. Kalnay; *Operational Ensemble Prediction at the National Meteorological Center: Practical Aspects*. Weather and Forecasting, Volume 8, pages 379?398, 1993.

[5] J. Williams; *Using random forests to diagnose aviation turbulence*. Machine Learning, Volume 95, pages 51?70, 2013.

[6] https://www.ncdc.noaa.gov/data-access/model-data/model-datasets/global-forcast-system-gfs

[7] http://www.emc.ncep.noaa.gov/GFS/doc.php

[8] http://nomads.ncdc.noaa.gov/data.php?name=access

[9] http://www.roc.noaa.gov/WSR88D/About.aspx