Using A Decision Tree and Neural Net to Identify Severe Weather Radar Characteristics

by

Ron Holmes[1]

NWS State College PA

1. Introduction

Forecasters have long known how to identify severe weather radar characteristics based on the shape, orientation, and strength of radar echoes. Certain signatures are associated with different types of severe storms. For example, an isolated severe storm will often, by its nature, be isolated or perhaps exist ahead of a line of convection. It will have a high reflectivity core and sometimes a hook echo. The hook echo is caused by a redistribution of the precipitation due to storm updraft and rotation. It will also tend to be long-lived and move in a southeast direction due to pressure perturbations within the storm that favor convective development on its southeast flank. This development is enhanced by strong environmental storm relative helicity that results from a highly sheared environment. Large hail occurs within these storms resulting in high VIL values.

Pulse severe storms often occur in a highly buoyant, weakly sheared environment. They tend to move slowly in this weak flow. Since they develop in an environment of very high low-level heat and moisture their maximum reflectivity core is often higher than normal storms. The high reflectivity can also be attributed to water coated ice associated with large hail. This also produces high VIL readings.

Linear storms, associated with line echo wave patterns (LEWP) and derechos, also have high reflectivity cores and tend to move fast. These storms often form in environments with strong winds and shear and relatively high instability. They typically to do not get as tall as pulse storms.

Humans have long known that the severe storm characteristics outlined above are good predictors in determining whether a storm is severe or not. They can then use this information to issue warnings. A critical question is whether decision trees and artificial neural networks can accomplish the same task.

Decision Trees and Artificial Neural Networks are two artificial intelligence algorithms that can be used for classification. Decision Trees measure the information gain from various combinations of the predictors and choose splits based on nodes that have the highest information gain. Pruning of the tree is used to prevent over-fitting of the data. This pruning mechanism maximizes information gain by getting rid of nodes that do not contribute much to information gain. In effect, what is left behind are the most effective predictors in a given data set. Artificial Neural Networks also can be used to classify data by repetitively presenting the predictors and know classification to the network and then adjusting the

[1] Corresponding author: Ron Holmes, National Weather Service, State College PA. 16803. ron.holmes@noaa.gov

weights of each node based on the error that occurs between its predicted classification and the known classification. With enough iterations the network slowly adjusts its weights to minimize the error between predicted and observed data.

The task then is to determine if an artificial intelligence (AI) algorithm can classify the type of storm using radar parameters as the predictors. The main purpose of this paper to show that two methods, Decision Trees and Artificial Neural Networks, can classify severe weather employing radar characteristics with some modicum of skill.

2. Data Preparation

a. Quality Control and Identification of Outliers

The data for this project came from an original study on severe weather forecast skill and its relationship to storm type (Guillot, et., al. 2008). Before using the algorithms it was necessary to look at the training data for errors or misclassifications. The training data supplied was put together by humans who looked at radar data and made judgment calls on the storm type classification. There were four types of storms classified: Non-severe, Line, Pulse, and Isolated storms. Visual inspection of the data revealed suspected measurement errors in the data and possibly some misclassification of storms. For example, the data had many storms classified as pulse storms and severe convective lines but with very low (< 10) max VIL values and low (<50) max reflectivity. There was also quite a number of missing data elements. In order to determine gross outliers in the data set the min, max, mean, and standard deviation (STD) was computed for each predictor for both the supplied training and test sets. Since our algorithms would be judged on performance on the test data set it was necessary to get the mean and STD for each predictor of the training data set as close as possible to those in the supplied test data set. This identified a few outliers (values > 2 STD's) in the training data set and those instances (4) were eliminated. A more thorough scrutinization of the training data by a group of meteorologists looking at the radar images would most likely result in a better classification of the storms. Although the training data had some suspected misclassifications those misclassifications were not changed because it was thought that the test data set would also have the same misclassifications in it. Also, without having the original radar data to look at it would be difficult to try to change the storm type in the training data for those suspected instances. Lastly, all missing values were changed to zero and the Lat and Lon of Centroid values for all instances were eliminated.

b. Training and Test Sets

Once the training data set was slightly modified programs were written to split the data into a training set and a test set. The file was read into memory and all instances were randomized in order to stratify the data properly. Before randomizing the data, a count was made of the number of non-severe, isolated, line, and pulse storms. It was noted that the number of instances for each storm type were not equal. For example, there were more non-severe and pulse storms than isolated and line storms. A better

training data set would have an equal number of instances for each class. Therefore, it was necessary to represent the same ratio for each class of storm in both training and test files so that the algorithm would not be biased. The overall ratio used was 70% for training and 30% for testing. Therefore each class had a different number of instances in the training and test files but the same proportion of 70% and 30% was kept for the training and test sets, respectively. Table 1 lists the breakdown used for the testing and training files. All instances in the training and test files were randomly stratified so that each class was randomly distributed.

Storm Type	Total number in original file	Number in training file based on 70% of original file	Number in testing file based on 30% of original file
Non-Svr	526	368	158
Isold	222	156	66
Line	208	146	62
Pulse	400	280	120

Table 1. Randomly stratified number of instances for the training and test files

In addition the training and test files were formatted for the decision tree and neural net algorithms.

3. Attribute Selection

a. Histograms

Attribute selection is an important part of the process to identify inputs to any artificial intelligence algorithm. If good predictors are not chosen performance will be poor both on the test set and in reality. Knowledge of the data for each class it represents and the upper and lower bounds of that data help determine what is good and what is suspect. There were suspected good predictors in this data set and some predictors that were suspected to not be as good due to difficulty in measuring or amount of missing data. A method to identify good predictors and attributes is to use histograms and a decision tree.

Histograms show the range and frequency of data and are a quick way to graphically see differences between classes for a particular attribute. The decision tree, through its process of measuring and maximizing information gain, will prune away poor predictors with the most influential predictors appearing at the top of the tree. Figure 1 shows histograms for each attribute, color-coded by storm type, for the original training set[2].

Non-severe storms are dark blue, Isolated storms are in red, Line storms in light blue, and Pulse storms in blue-green. The Max Reflectivity, Max VIL, and Mean Reflectivity immediately stand out as good predictors because the frequency distributions for each class appear to be well separated. Other good predictors are MESH/LifeTimeMESH, Aspect Ratio, and Rotation. Intuitively, and meteorologically, these predictors make sense. However there are subtle differences for other predictors than cannot be gleaned from a visual histogram. To further explore and measure the contribution of these predictors the data set was run through a decision tree.

b. Decision Tree Analysis of Attributes

[2] These images are best viewed by using a higher zoom factor in your document reader.

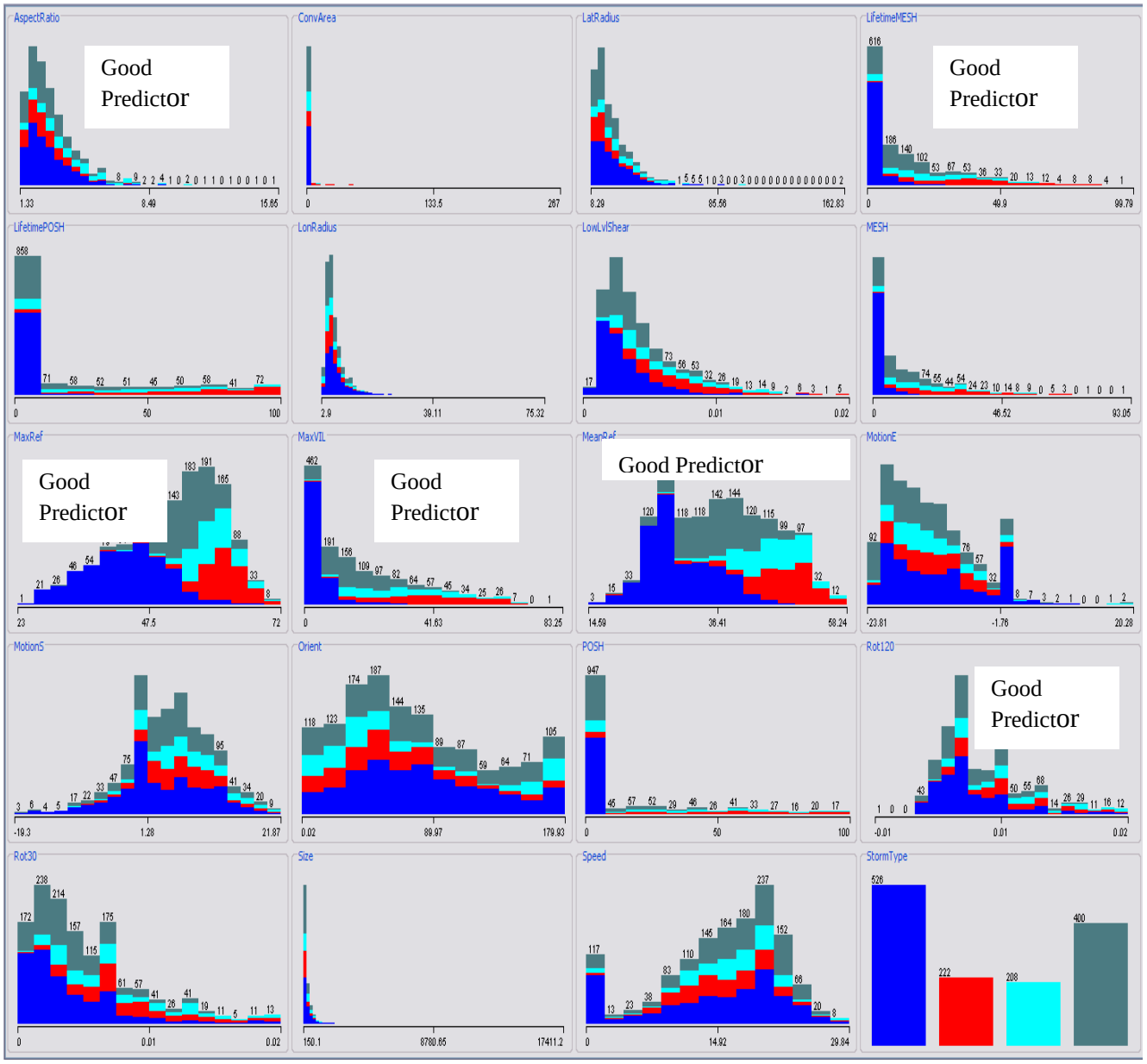

Figure 1. Frequency histograms for all attributes in the original training set for each class. Non-severe storms are dark blue, Isolated storms are in red, Line storms in light blue, and Pulse storms in blue-green. The lower right histogram shows each class and number of instances for each class.

The decision tree used was a version of Quinlan's (1990) C4.5 algorithm available in the Wakaito for Environment for Knowledge Analysis (Weka) tool (Witten and Frank, 2005). When running the training data through the decision tree a large tree is produced with many nodes or decision points. This results in a 66.8% of correctly classified instances. However such a large tree is likely to be overfit. Weka has a number of tunable parameters that allow for pruning of the tree to reduce overfitting such as the confidence factor and minimum number of instances per leaf node. Using a confidence factor of 0.5 and a minimum number of objects per leaf of 10 results in smaller tree (19 leaves) with a higher percentage of correctly classified instances (69.5%). The resulting tree is shown in Figure 2.

The decision tree shows the attributes with the most information gain at the top of the tree with lesser influential attributes towards the bottom. Meteorologically, it makes sense to see Max VIL, Max Reflectivity, and Mean Reflectivity, MESH, and Aspect Ratio at the top of the tree. These predictors were shown to be well separated in the histograms. However the decision tree process ferrets out more subtle details and relationships in the data that would be hard for a human to judge by looking at the histograms alone. These subtle differences appear farther down the tree. Note that the low values of Max VIL used in the decision split are likely the result of bad measurements by the human classifying the data set. Better quality control of the data would likely result in a more realistic tree that made sense meteorologically, at least with regards to the Max VIL node. It would be unwise to use this tree in a real-time operational setting. Nevertheless, even with suspected bad data, the most prevalent predictors rise to the top of the tree. In addition, sensitivity tests were done by removing particular attributes, one at a time, from the data set and running it through the decision tree each time to measure the decrease (or increase) in predictive skill. The attributes that lowered the skill after having been removed were deemed important and added to a list of possible attributes to keep. Others that did not have an influence were eliminated. The tree and sensitivity tests suggested the number of predictors for radar classification should be reduced from the original 19 to 11. These are identified in the nodes of the tree.

c. Confirmation of Decision Tree Analysis

Another aspect one hopes to find in the results of the decision tree is that the tree makes sense based on what meteorologists already know about severe weather radar signatures outlined in the introduction. This tree does appear to support those ideas. Each leaf node contains the number of correctly classified events and incorrectly classified events. For example, the left side of the tree is dominated by the non-severe events with very low Max VIL, low Max Reflectivity, and low probability of hail. There were some pulse storms on this side of the tree but the majority of the pulse events appear on the right side where VIL, Reflectivity, and MESH are higher. The Aspect Ratio is a very good predictor for determining line severe storms quickly identifying the majority (64) very high in the tree. Descending further in the tree shows more subtle differences that begin to flush out rotational aspects of severe storms. Weak low level shear and slowly moving storms separate the pulse events from additional line events. The majority of the isolated severe storms occur, as one would expect, where low level shear is high. These storms are developing in an environment which supports favorable horizontal vorticity which gets tilted vertically by updrafts and results in rotation. Such storms are highly organized and usually long-lived. The resulting strong updrafts support large hail which is identified by the Lifetime POSH and Rotation decision nodes.

4. Identifying Non-Linear Relationships with a Neural Network

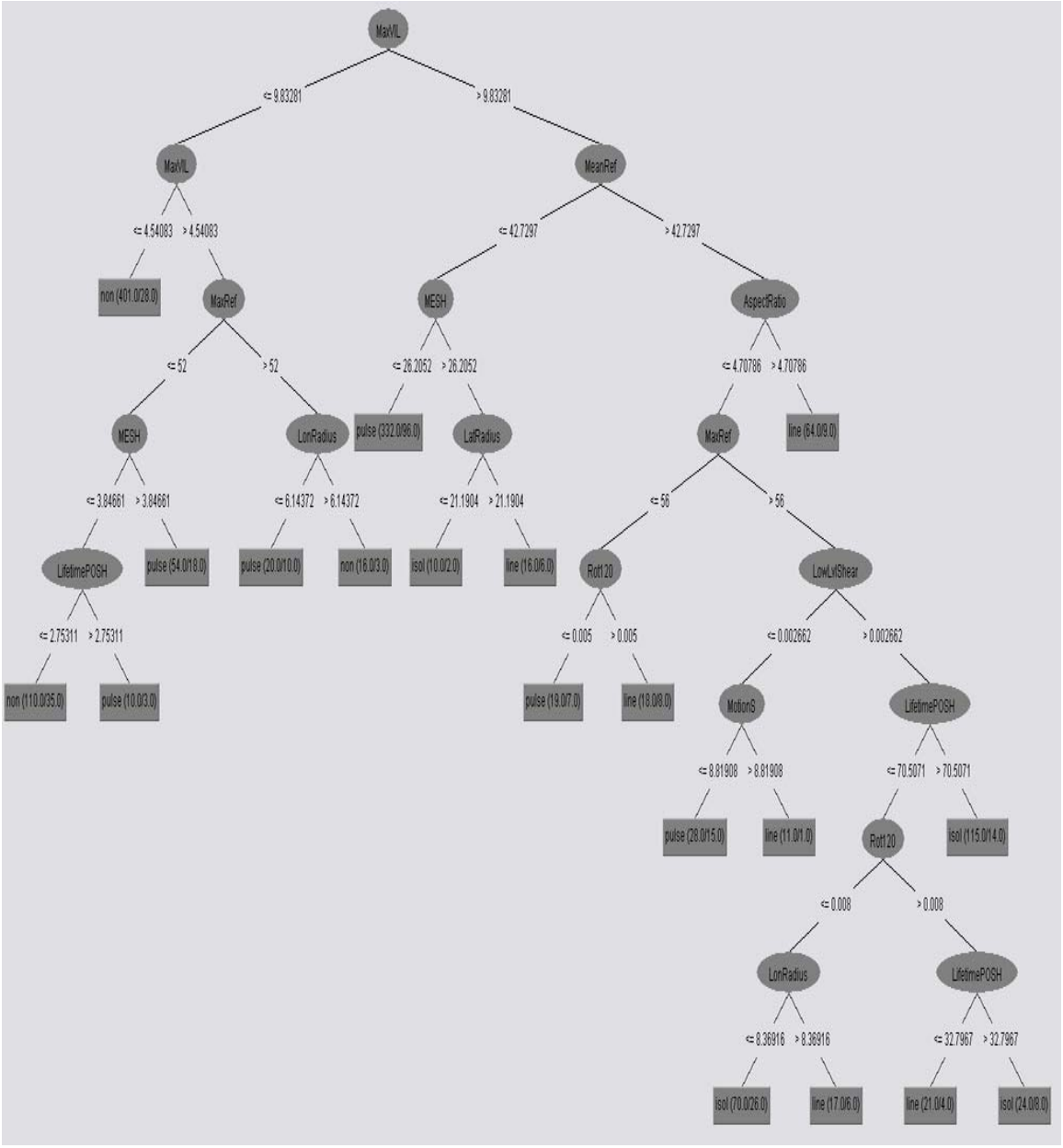

Figure 2. Decision Tree for radar classification data using a Confidence Factor or 0.5 and a minimum number of instances per leaf of 10.

a. Visual Inspection of Scatter Plots

There are very few processes in meteorology that are linear. The atmosphere is a three-dimensional, chaotic fluid. Thus, it should be no surprise that using linear models to identify non-linear relationships in meteorological data would not do well. This radar data set is no exception. Figures 3, 4, and 5 show a few examples of non-linear relationships in this radar data. Each instance of the data is plotted and color-coded for the type of storm. Non-severe storms are dark blue, pulse storms are light blue, line storms are green, and isolated storms are red

b. Artificial Neural Networks

A decision tree can identify non-linear relationships but a more adept algorithm

and potentially better one is the application of a Neural Network (NN). Reed and Marks (1999) showed that a two layer neural net can model any non-linear function. Weka does include a neural network interface however it is difficult to use because a large number of parameters (nodes, learning rate, momentum, etc) must be manually adjusted for each training run via a GUI interface. Therefore a neural network was developed in Java so that different network architectures could be automated and tried on the training and test sets. The Java implementation has a number of the adjustable parameters as the Weka version. It is a two-layer, back-propagation network with adjustable number of nodes in each hidden layer, learning rate, momentum, and weight decay. It also uses the logistic sigmoid as its activation function for all hidden nodes and includes a bias node for the input and hidden layers. Different network structures were run on the training data set and the test data set was used to measure the skill of the network. The training and test files for the neural net were made by splitting the original data set into a 70% train and 30% test configuration after randomly stratifying all instances in the original file. The relative distributions of predictands for the taining and test files were proportioned in the same manner as was done for the decision tree. All inputs to the neural network were normalized (for both training and test sets) using the following equation according to Witten and Frank (2005).

Normalized Value = (Value-min)/(max-min)

where min and max represent the minimum and maximum values found for each particular predictor. For example, each input value to the neural net for the Max Reflectivity was normalized by using the min and max of 23 and 72, respectively.

	Training TSS: 0.67		
	POD	FAR	CSI
Non Svr	0.86	0.14	0.75
Isold	0.75	0.28	0.57
Line	0.56	0.25	0.46
Pulse	0.75	0.31	0.56

Table 2. Results of Neural Network on training data showing True Skill Score (TSS), Probability of Detection (POD), False Alarm Rate (FAR), and Critical Success Index (CSI) for each storm type.

Various skill scores were computed based on the predicted and observed outcome for both training and test data sets for each run of the network. Since the competition would be judged on the highest True Skill Score (TSS) it was decided to use this skill score as a measure of forecast robustness in the Java implementation.
The goal was to
produce the best network that achieved the highest, but nearly equal, TSS for both the training set and testing set. The TSS for training and testing should be as close as possible. Anything else is overfit. The best network achieved a TSS of 0.67 on the training data and 0.70

	Testing TSS: 0.70		
	POD	FAR	CSI
Non Svr	0.88	0.12	0.78
Isold	0.71	0.27	0.55
Line	0.51	0.25	0.43
Pulse	0.80	0.29	0.60

Table 3. As in Table 2 except for testing data.

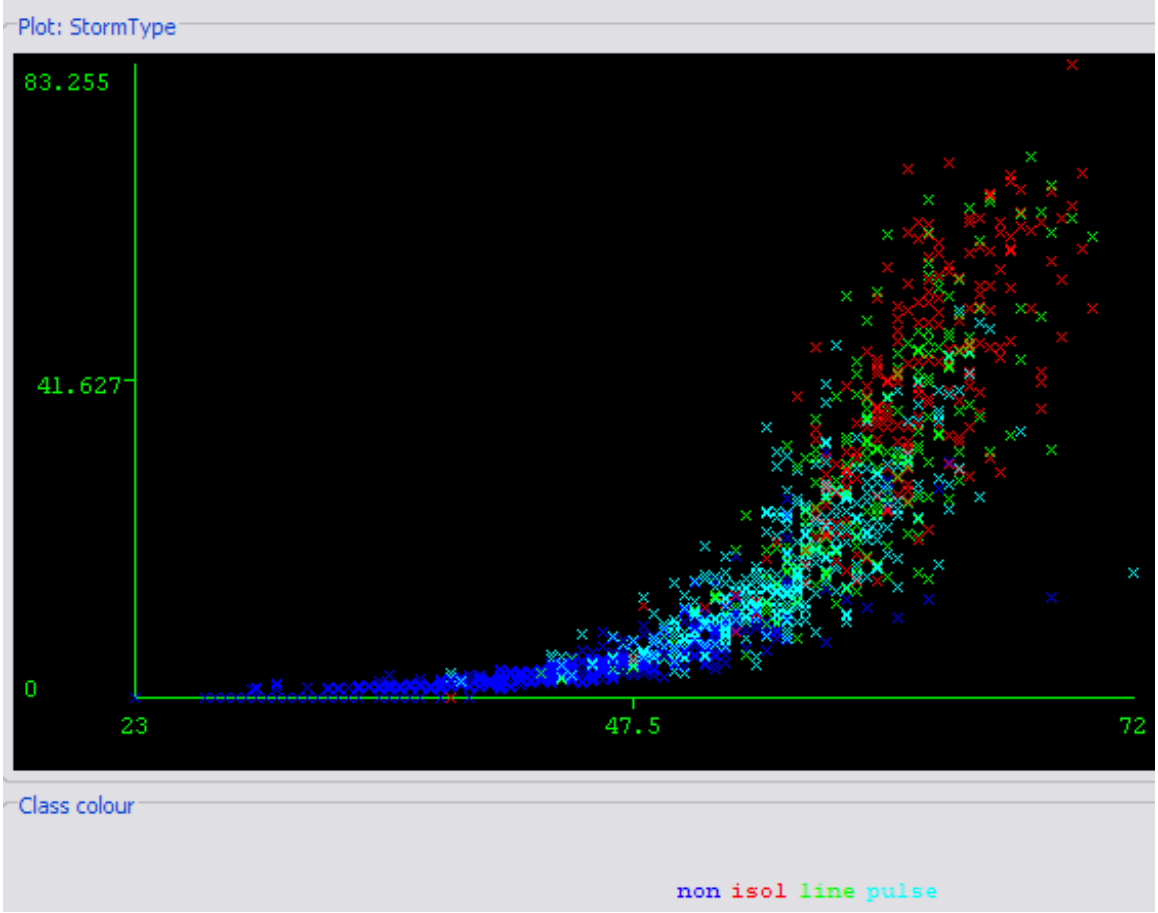

Figure 3. Graph of Max Reflectivity (X axis) vs. Max VIL (Y axis) for each storm type. Non-severe storms are dark blue, pulse storms are light blue, line storms are green, and isolated storms are red.

on the test data. In the original study (Guillot, et. al., 2008) that used a decision tree the TSS was 0.71 for training and 0.58 for testing. Since the TSS in their study was much lower for the test set than the training set it indicated that their model was slightly overfit. In this study, although the training TSS was slightly lower than the test TSS, it was felt that since they were so close the network indicated a good model that would be robust enough to generalize well on new, unseen data. Higher training TSS's were easily achieved with more complex networks but the accompanying testing TSS scores fell much lower indicating these networks were overfit. Since the test TSS in this study was higher than the test TSS in the original study it was thought that this model was an improvement and competitive enough to enter the competition.

c. Results of Neural Network

The results of the neural network are show in Tables 2 and 3. It can be seen that the POD, FAR, and CSI between the training and testing data sets are close indicating an unbiased and properly fit network. The Line type severe storms were the hardest to identify. This could be due to the fact that the number of instances in the original data set was the lowest of all 4 types and not properly represented. The CSI for both training

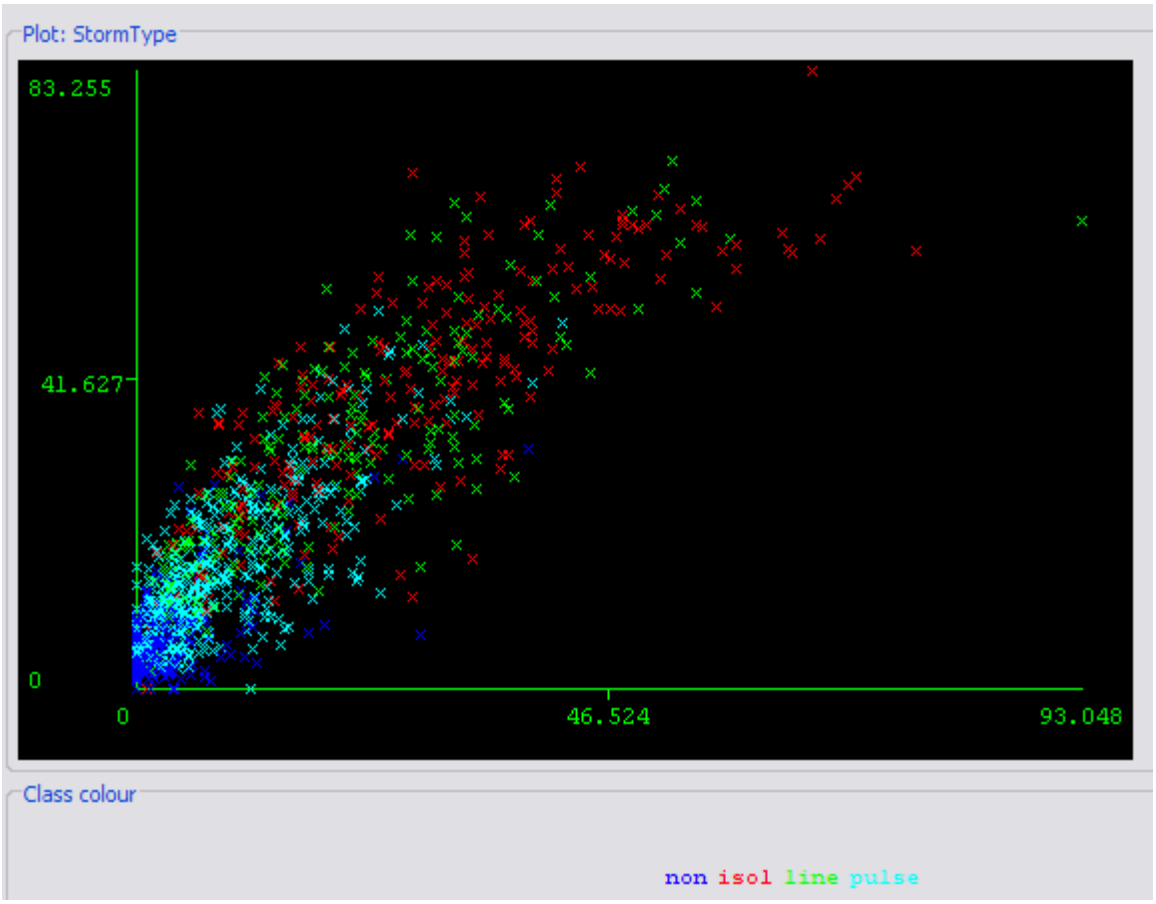

Figure 4. As in Fig 3 except graph of MESH (X axis) vs. Max VIL (Yaxis)

and test sets are improved over the CSI in the original study indicating the neural network approach is an improved model over the decision tree approach alone.

Tables 4 and 5 show output from the neural network in confusion matrix form for the training and test data sets.

For the training confusion matrix shown in Table 4 the numbers are scaled by the number of iterations performed by the network. This was done in order to keep the output results in a compact form and for visual representation only. This confusion matrix was a result of a network with 600 iterations, 20 hidden nodes in layer 1, 10 hidden nodes in layer 2, a learning rate of 0.08 and momentum of 0.3. Therefore the true number of hits, misses, etc are those in the table multiplied by 600. Since the test data set was only run through the network once no adjustment is necessary. The diagonal values highlighted in red indicate pure hits and the Accuracy values of each row identify the POD for each class. The Accuracy along the bottom indicates that all storms forecast by the model of that class are truly made up of that class.

5. Conclusion

a. Improvements over Original Study

A decision tree and neural net were used to explore relationships in radar data used to characterize severe storms. The decision tree alone showed comparative skill to the original study in correctly identifying severe storms. Its use in this

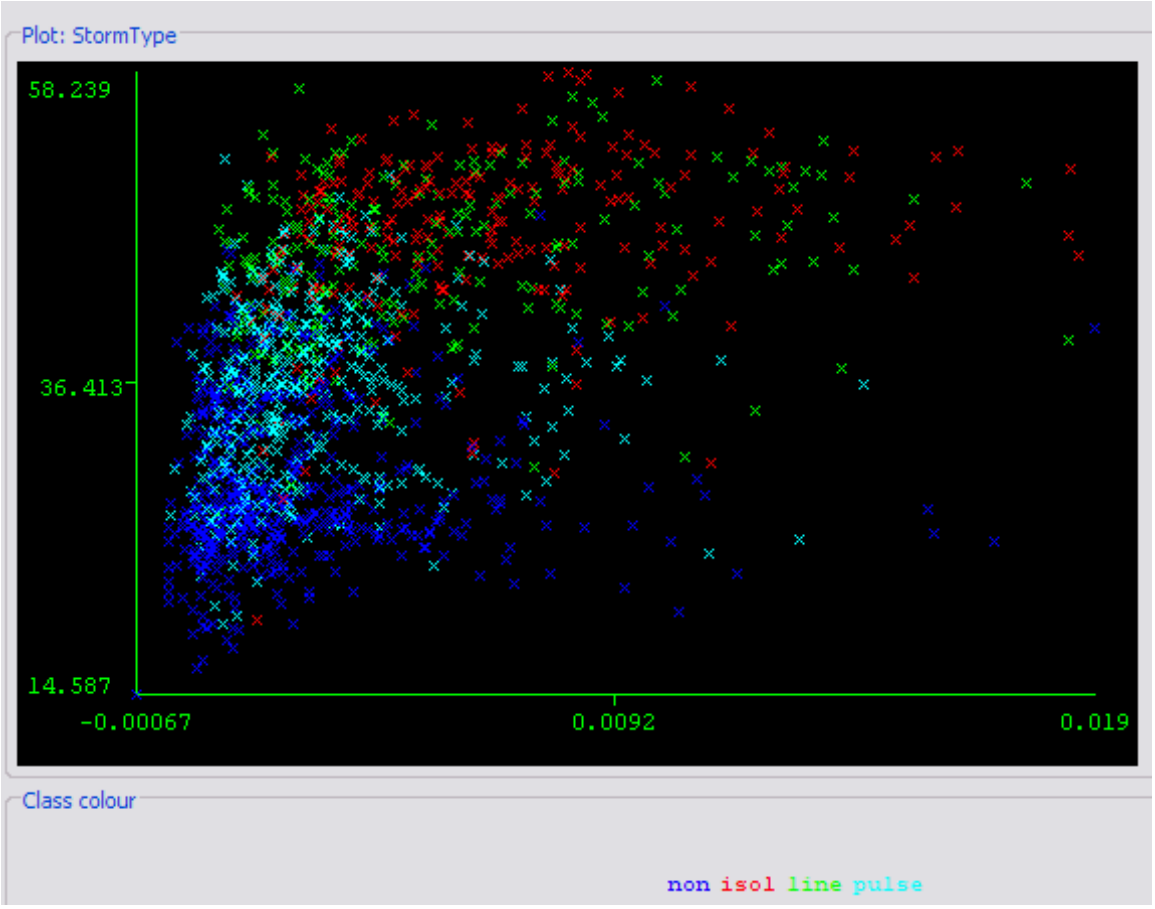

Figure 5. As in Fig 3 except for graph of Low Level Shear (X axis) vs. Mean Reflectivity (Y axis)

study was to help identify the more influential attributes for severe storm classification and to confirm attributes that have always thought to be important by meteorologists in identifying severe weather characteristics of radar data. Sensitivity tests and pruning helped flush out important predictors and reduced the number of inputs for the neural net providing for faster training and reduced overfitting.

The choice of a neural network over the decision tree for classification was due to the non-linear relationships thought to exist in the radar data. A two layer neural net can model any non-linear function. The results from this implementation of a neural net showed increased skill over the decision tree approach based on a higher CSI for both training and test data over the original study. The higher CSI scores were mainly due to lower FAR values than in the original study. When comparing Accuracy scores of the confusion matrix between the original study and this one it was found to be equal in skill for the training set. However the confusion matrix for the test set in this study showed increased Accuracy scores (as well as TSS scores) indicating a forecast model that was more robust and better at generalization than in the original study. It was also found that forecast skill of the neural net was highly dependant on the number of training instances representing each predictand. Those predictands that had a lot of training instances (Non-Svr and Pulse storms)

had better skill scores than predictands with relatively fewer training instances (Line storms). An equal distribution of training instances for each predictand would likely produce skill scores that were relatively even.

b. Future Applications

The neural network approach to identifying severe radar characteristics shows good promise for the future. It would be easy to implement in an operational setting and would classify storms in a matter of seconds given the number of inputs for which it was trained. Providing that input and formatting it for the neural net would be the difficult part. One could possibly do this by setting up a web cam trained on a loop of radar imagery and capturing the images. Software would be needed to take the input images and translate them into some coded, perhaps binary, representation. Clustering analysis could be performed to separate the echoes and then the attributes (Max VIL, Max Reflectivity, Aspect Ratio, etc) could be recorded after each volume scan for each cluster of echoes. These attributes would be fed into the neural network for classification. Over time a probability distribution could be built from successive classifications with each volume scan. The forecaster could then be alerted for the probability for each class of severe storm on the radar. This automated classification scheme would be very important in the future when the phased array radar becomes operational. With such large amounts of data arriving in one-minute resolution volume scans it would be very difficult for humans to keep up with the increased data flow by using manual identification techniques.

6. REFERENCES

Guillot, E., M., T. M. Smith, V.Lakshmanan, K. L .Elmore, D.W. Burgess, and G. J. Stumpf, 2008: Tornado and Severe Thunderstorm Warning Forecast Skill and its Relationship to Storm Type. American Meteorological Society, 24th Conference on IIPS, New Orleans, LA, Jan. 2008

Quinlan, J. R., 1990: Induction of Decision Trees. Readings in Machine Learning. Morgan Kaufmann, 500 Sansome St., Suite 400, San Fransisco, CA 94111

Reed, Russell D., and R. J. Marks, 1999: Neural Smithing. Supervised Learning in Feedforward Artificial Neural Networks. The Massachusetts Institute of Technology

Witten, Ian H. and E. Frank, 2005. Data Mining. Practical Machine Learning Tools and Techniques. Morgan Kauffman, 500 Sansome St., Suite 400, San Fransisco, CA 94111

Training Confusion Matrix TSS: 0.67

		Forecast Class				
		Not Svr	Isold	Line	Pulse	Accuracy
Observed Class	Not Svr	318.4	1.9	0.9	47.8	86.3%
	Isold	2.0	117.0	17.8	19.2	75.0%
	Line	5.0	30.3	81.9	28.8	56.1%
	Pulse	45.9	14.1	9.8	211.2	75.2%
	Accuracy	85.8%	71.7%	74.2%	68.8%	

Table 4. Confusion Matrix for training data for each severe storm class. Numbers along the diagonal indicate correctly identified instances.

Testing Confusion Matrix TSS 0.70

		Forecast Class				
		Not Svr	Isold	Line	Pulse	Accuracy
Observed Class	Not Svr	139.0	0.0	0.0	18.0	88.5%
	Isold	1.0	47.0	4.0	14.0	71.2%
	Line	4.0	17.0	32.0	9.0	51.6%
	Pulse	15.0	1.0	7.0	96.0	80.7%
	Accuracy	87.4%	72.3%	74.4%	70.1%	87.4%

Table 5. As in Table 3 except Confusion Matrix for testing data.