

J5.2 A Neural Network for Detecting and Diagnosing Tornadic Circulations using the Mesocyclone Detection and Near Storm Environment Algorithms

V Lakshmanan^{1,2}, Gregory J. Stumpf^{1,3}, Arthur Witt^{2*}

Abstract

A Mesocyclone Detection Algorithm (MDA) and a near-storm environment (NSE) algorithm have been developed at the National Severe Storms Laboratory. The MDA analyzes azimuthal shear in Doppler velocity data in 3 dimensions to identify storm-scale circulations. Sometimes, though not always, these circulations are precursors to tornadoes. Marzban and Stumpf (1996) developed a neural network based on the MDA parameters to identify which of the circulations would be tornadic using a small set of data cases. That work was extended to cover 43 storm days in Marzban (2000) using a more robust methodology.

In this paper, we further extend the work to use 83 storm days and introduce some variations that improve neural network performance over that achieved by Stumpf and Marzban (2002). We also evaluate whether the incorporation of near-storm environment (NSE) data from those days can improve the predictive capability of the neural network.

On an independent test set of 27 storm days, we achieve a Heidke Skill Score (HSS) of 0.41 using just the MDA parameters and a HSS of 0.45 using a combination of MDA and NSE parameters. The Critical Success Index (CSI) for the MDA-only neural network is 0.29, while the CSI for the neural network with both MDA and NSE parameters is 0.32.

1. Introduction

A classification neural network can be thought of as a non-linear function that maps real-valued input variables to a number varying between 0 and 1. The process of “training” a network is the process of finding the optimal non-linear function of a given form that provides values closest to the target values (0 and 1) for a set of inputs called the training set. In neural network training, finding global optima is usually not necessary; a sufficiently deep local minimum often suffices (Bishop 1995).

For a three-layer perceptron network, the most commonly used form of neural network, the mapping function is:

$$f(x_1, \dots, x_n) = g(\sum_j w_j h(\sum_i w_{ij} x_i)) \quad (1)$$

*¹The Cooperative Institute of Mesoscale Meteorological Studies (CIMMS), University of Oklahoma, ²The National Severe Storms Laboratory, Norman, OK. ³Meteorological Development Laboratory, National Weather Service, Silver Spring, MD

where $x_1, x_2 \dots x_n$ are the inputs to the network and x_0 is a bias term that will be optimized in the training process along with the weights w_j and w_{ij} . The function g is called the output node activation function and the function h is called the hidden node activation function.

Neural networks have found wide applicability in meteorology, from rainfall prediction (Venkatesan et al. 1997) to tornado diagnosis (Marzban and Stumpf 1996) and quality-control (Lakshmanan et al. 2003).

The National Severe Storms Laboratory (NSSL) Mesocyclone Detection Algorithm (MDA) (Stumpf et al. 1998) is designed to detect a wide variety of circulations of varying size and strength by analyzing the radial velocity data from a Doppler weather radar. The near-storm environment (NSE) algorithm uses analysis grids from the Rapid Update Cycle (RUC) model that is produced hourly to derive features such as the 0 - 3km storm relative helicity and surface-based CAPE. A set of 245 attributes was considered in this study.

1a. Scalar Performance Measures

The measures reported in this paper are the probability of detection (POD), False Alarm Ratio (FAR), Critical Success Index (CSI) (Donaldson et al. 1975) and the Heidke Skill Score (HSS) (Heidke 1926). All these are computed on the confusion table:

	Identified as	
	non-tornadic	tornadic
Non-tornadic	<i>null</i>	<i>fa</i>
Tornadic	<i>miss</i>	<i>hit</i>

(2)

where *hit* and *null* are tornadic and non-tornadic mesocyclones respectively that were correctly identified by the neural network. *fa* are mesocyclones classified by the neural network (NN) as being tornadic, but verified as being non-tornadic, while *miss* are tornadic mesocyclones mis-classified by the neural network as being non-tornadic. The four measures of skill are:

$$\begin{aligned}
 POD &= \frac{hit}{hit+miss} \\
 FAR &= \frac{fa}{hit+fa} \\
 CSI &= \frac{hit}{hit+miss+fa} \\
 HSS &= \frac{2*(null*hit-miss*fa)}{(fa+hit)*(fa+null)+(null+miss)*(miss+hit)}
 \end{aligned}
 \tag{3}$$

where *POD* is the probability of detection and *FAR* the false alarm ratio. The Critical Success Index, *CSI* (Donaldson et al. 1975), doesn't take into account non-tornadic mesocyclones that are correctly classified, and is therefore a very conservative estimate of skill. The Heidke Skill Score, *HSS* (Heidke 1926), accounts for the relative a-priori distribution of tornadic and non-tornadic mesocyclones. Measures commonly used in pattern recognition or data mining (such as the classification rate) tend to exaggerate skill in rare-event situations and are therefore not reported here.

1b. ROC curves

In the case of neural network training, where the outputs are actually real-valued numbers in the range 0 to 1, it is necessary to threshold that output before computing a confusion table. There is value, however, in the probabilistic output of the neural network itself. So, the complete performance at all thresholds is shown using a receiver operating characteristic (ROC) plot (Masters 1993). See Figure 3 for an example of a ROC plot.

Classifiers with curves above the diagonal shown in the plots can be considered skilled. The closer a classifier is squashed to the left and top boundaries of the graph (i.e. the larger the area between the curve and the diagonal), the better it is. In the ROC plots in this paper, three different types of curves are shown. The curves labeled

0,1,2, etc. show performance of the network on different slices of the training data set. The closer together these curves are, the less the variability of the network's performance when presented with different data. The curves labeled "train" show the performance of the network on the entire training data set. The curves labeled "test" show the performance of the network on an independent set of testing data. On each ROC curve, the performance at three different thresholds are shown as a,b,c. The actual values of these thresholds are not relevant, but the more clustered these points are, the better the generalization of the network when subjected to warning thresholds.

2. MDA parameters neural network

We used a fully feedforward resilient backpropagation neural network (RPROP, Riedmiller and Braun (1993)) with one hidden layer. The hidden node activation function was the tanh function, while the output node activation function was the logistic function $1/(1 + \exp(-x))$. The error function that was minimized was a weighted sum of the cross-entropy (which Bishop (1995) suggests is the best measure of error in binary classification problems) and the squared sum of all the weights in the network:

$$E = E_e + \lambda(\Sigma w_{ij}^2 + \Sigma w_j^2) \quad (4)$$

The cross-entropy error E_e is of the form suggested by Bishop (1995) and is defined as:

$$E_e = - \sum_{n=1}^N (t^n \ln y^n + (1 - t^n) \ln(1 - y^n)) \quad (5)$$

where t^n is the target value of the nth training pattern (0 if non-tornadic and 1 if tornadic) while y^n is the actual output of the neural network for that pattern input. N is the total number of patterns.

The second, square weights, term attempts to reduce the size of the weights, and thus improves generalization (Krogh and Hertz 1992). The relative weight, λ , of the two measures is computed every 50 epochs within a Bayesian framework.

The NSSL MDA was run on 83 cases in which ground truth of tornadoes exists. The output contains information from every strength and size of 3D circulation. Tornado ground truth is based on temporal and spatial proximity. If there was a tornado reported between the beginning and ending of the volume scan, and the report was within reasonable distance of a circulation detection (input manually), then the ground truth value was given as 2. If a circulation detection fell within the prediction "time window" of -20 to +6 minutes of the ground truth report duration, then the ground truth value is given a 1. All other circulations were given as 0 to correspond to no ground truth. Tornadoes not detected by the MDA are unscored. For more details on the MDA or the truthing process, please see Stumpf et al. (1998). The complete list of 245 features or attributes as well as the data containing truthed detections from the MDA are available online¹.

2a. Training Method

These were the steps taken to train the network on MDA parameters:

1. Extract out MDA detections that were truthed. In this study, both detections (truth value of 2) and predictions (truth value of 1) were treated identically, i.e. the mesocyclone was assumed to be tornadic. Detections that fell over a large body of water or were of MDA rank 0 were not included in the truthing process, and are therefore not included in the training or in the statistics.

¹<ftp://ftp.nssl.noaa.gov/users/lakshman/mdanse.zip> (69MB)

2. The input features were all normalized so as to have zero mean and unit variance over the entire data set.
3. Thirteen of the features were identified through prior knowledge as having a univariate tendency. For example, it is known that high rotational velocities at lower levels are required for circulations to be tornadic. The rotational velocity threshold below which the apriori probability of tornadoes fell below 0.05 was applied, and all MDA detections below this threshold were pruned. This serves to greatly limit the training data set presented to the neural network training. In the validation and independent testing, the same thresholds are applied; quoted results include the effect of this prior processing.
4. For every parameter, a reasonable range was established and values outside this range were treated as being missing and replaced by the data set mean.
5. The training data were divided in the ratio 7:3. The data were divided case-wise i.e. for every 3 storm days retained for validation, 7 storm days were used for training. Where independent testing was desired, the entire data set was divided in the ratio 46:20:34 with 34% of the cases used for independent testing.
6. The division was repeated several times (“bootstrapping”) and the neural network trained on the training data set. In each training run, the number of hidden nodes was varied from 1 to $2+N$ where N is the number of input features. At each setting of the hidden node, five training runs with different random seeds were performed. The training run was stopped when the validation cross-entropy started to increase (“early stopping”). The architecture (number of hidden nodes, and weights w_{ij}) at which the globally minimum cross-entropy was obtained on a validation set was selected as the final network.
7. The “warning threshold” was chosen as the threshold at which the maximum Heidke Skill Score (Heidke 1926) was achieved on the validation set. This threshold was applied to the output of the network when processing the independent test set to report performance measures.

2b. Comparison with Marzban and Stumpf

The training regimen that we followed for the neural network differs somewhat from that of Marzban (2000); Stumpf and Marzban (2002). The main points of difference are these:

1. The error criterion minimized by us is a weighted sum of the cross-entropy and the sum of the square of weights. The earlier study did not do any weight decay. Weight decay (Krogh and Hertz 1992) can decrease over-fitting since smaller weights tend to generalize better.
2. Our error minimization method employed resilient propagation while the the earlier study employed a scaled conjugate gradient (SCG) followed by simulated annealing. Recent studies (Caruana et al. 2001) show that SCG methods can over-fit where backpropagation doesn't.
3. We divided the training data case-wise whereas the earlier study divided the training data pattern-wise, i.e. Marzban (2000); Stumpf and Marzban (2002) put all the MDA detections into a pot and divided it randomly whereas we divide up the storm days (not the detections) randomly. We believe that our method is better because the MDA detections from a single day (consider consecutive volume scans as an extreme example) tend to be highly correlated. Thus, if we divide pattern-wise, the validation set could be correlated in parts with the training set leading to a spurious global minimum.
4. We pruned input features and outliers automatically using the apriori threshold, rather than manually as was done by Stumpf and Marzban (2002). This was done essentially to save time since there are more than 200 NSE parameters whereas the earlier study had only the MDA features.

Method	POD	FAR	CSI	HSS
Marzban, Stumpf	0.36	0.69	0.20	0.29
This paper	0.34	0.38	0.28	0.38

Table 1: Comparison of performance of networks trained using the same 29 cases and tested on the same 14 cases used by Marzban (2000). See Section 1a for an explanation of these measures.

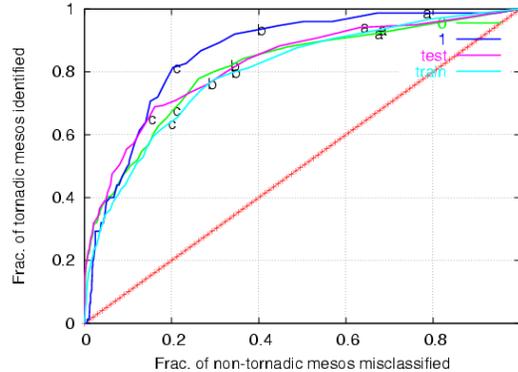


Figure 1: A ROC curve showing the performance of the neural network on two bootstrap runs' validation sets, on the entire training set and on the testing data set. See Section 1b for an explanation of ROC plots.

Since we are following a different training regimen, we believed it worthwhile to test out our training regimen on the same data set used by (Marzban 2000; Stumpf and Marzban 2002). In that data set, consisting of 43 cases, 29 cases were used for training using cross-validation and the remaining 14 cases were used as an independent test set. The only parameters available were from the MDA – the NSE inputs were not available for all 43 cases in (Marzban 2000; Stumpf and Marzban 2002). It is the performance of the trained network on the independent testing set that we report in Table 1.

The receiver operating characteristic (ROC) plot that depicts performance of the neural network developed using the techniques described in this paper is shown in Figure 1. Notice that the training, validation and test sets all show approximately the same performance. Although Stumpf and Marzban (2002) obtained performances similar to ours on their validation set, their test case performance was somewhat worse, due to poorer generalization. In other words, the changes we made to the training regimen – employing weight decay, resilient propagation and achieving a less correlated validation set – did result in a more generalized neural network.

2c. Performance on 83-case set

The 43-case data set used in the earlier studies was comprised almost entirely of supercell storms. The 83-case set, on the other hand, has a number of isolated events and weak tornadoes. We thus needed to make sure that a neural network was capable of generalizing across this data set. We manually divided the 83 cases into two pieces – a 56-case training set and 27-case independent testing set. The division was done subjectively so that both sets covered the spectrum of tornadoes shown in the entire 83-case set. Bootstrapping was done by dividing the 56-case set and the resulting networks were evaluated on the independent 27-case set. This served to confirm for us that the resulting networks were sufficiently general.

Data Set	POD	FAR	CSI	HSS
27-independent	0.44	0.53	0.29	0.41
Best Validation	0.42	0.51	0.29	0.40

Table 2: Performance of networks trained using the MDA parameters. The first line is the performance of a network trained on 56 cases tested on an independent set of 27. The second line describes the performance of the final network, trained using all 83 cases and chosen amongst all the bootstrapping runs by having the best validation cross-entropy.

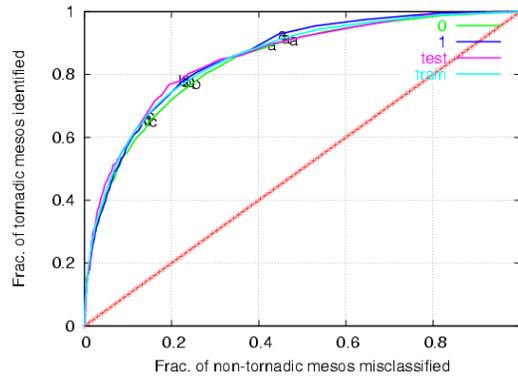


Figure 2: ROC curve showing the performance of a network trained with 56 cases and tested on an independent set of 27.

At this point, the same training regimen was used to bootstrap the entire 83-case set. This network is what is used operationally. It should be noted, however, that the ROC curve for this final network does not have any truly independent testing cases – the best that can be shown is the variability of the network on different slices of the 83-case set. The various performance measures are shown in Table 2 and Figures 2 and 3.

3. MDA + NSE neural network

The statistics of the data set change dramatically when NSE parameters are added as inputs to the neural network. Intuitively, there are two main reasons for this:

1. The number of inputs is about ten times more. Because the number of weights increases as a square of the number of inputs in a single hidden layer neural network, we would actually have 100 times as many free variables in the network training. Thus, a data set that is more than enough for training a MDA network might prove deficient in training a MDA+NSE network.
2. NSE parameters (much more so than MDA parameters) are highly correlated across time. Detections from the same time period will all tend to differ only slightly in their NSE variables. Thus, the variability of the input features is driven more by the number of data cases (83) than by the number of detections (approximately 23,000).

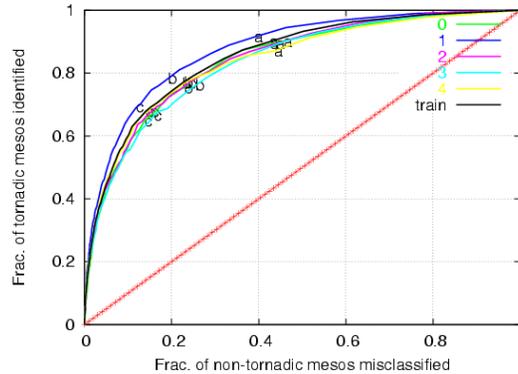


Figure 3: ROC curve showing the performance on various bootstrapped sets of a network trained with all 83 cases.

Thus, we decided to not use all the NSE parameters, but only a selected few. This was our strategy in selecting the parameters:

1. Choose based on meteorological understanding which NSE parameters were likely to have some effect on the tornado probability. This initial selection reduced the number of attributes considered from 245 to 76 (See Appendix for this list).
2. The Pearson's correlation coefficient was computed on each pair of features and in pairs whose correlation coefficient was greater than 0.8, one of the features was dropped. This 0.8 threshold was chosen arbitrarily.
3. All the features were considered one-by-one as univariate predictors i.e. the ability to predict based on just this variable whether the detection was tornadic or not. The top f (fraction) of the features was chosen as input to the network training.

To choose the ideal value of f , we again resorted to the validation set. We trained different networks using values of f ranging from 0.1 (only the top 10% of univariate predictors) to the top 1.0 (all the NSE features). Then, we chose the value f that yielded the minimum over-training based on the validation set. In Figure 4, the variation of the best cross-entropy per pattern on the training and validation sets as the fraction f is varied is shown. In general, the trend is for lower errors on both the training and validation sets as we increase the number of features. However, lower absolute errors does not necessary mean a better generalization capability. The difference between the training and validation error is one measure of over-fitting, and the variation of this difference with f is shown in Figure 5. Here, the trend is for the generalization error to decrease and reach a minimum at $f = 0.3$ and then to increase after that.

A value of $f = 0.3$ was thus chosen empirically, and the top 30% of features based on their univariate predictive capability were chosen as inputs to the MDA+NSE training process. It should be noted (See Table 3 and Figure 4) that the highest Heidke Skill Score and the lowest entropy are both obtained at $f = 0.9$. Thus, we are sacrificing performance for generalization by using just 30% of the inputs.

The various performance measures are shown in Tables 4 and 5 and Figures 6 and 7.

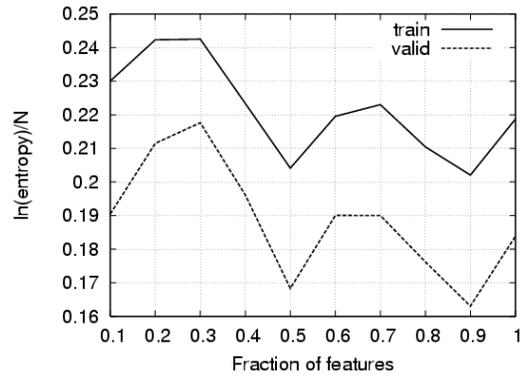


Figure 4: Variation of the neural network training and validation errors as the number of input features is increased. The bootstrapping run with the best validation error is shown.

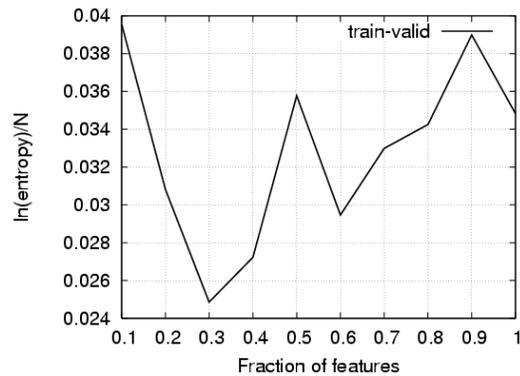


Figure 5: Difference between training and validation errors as the number of input features is increased. The bootstrapping run with the best validation error is shown. We chose to use $f = 0.3$ as it provides the best generalization performance.

Fraction	POD	FAR	CSI	HSS
0.1	0.45	0.58	0.28	0.37
0.2	0.45	0.59	0.27	0.37
0.3	0.51	0.61	0.28	0.37
0.4	0.51	0.50	0.34	0.45
0.5	0.50	0.41	0.37	0.50
0.6	0.54	0.53	0.34	0.45
0.7	0.34	0.38	0.28	0.40
0.8	0.49	0.46	0.35	0.47
0.9	0.58	0.48	0.38	0.50
1.0	0.50	0.49	0.34	0.46

Table 3: Variation of the skill scores computed on the best validation set with different pruning fractions f . The neural networks were trained on 56 cases using a fraction f of the MDA+NSE parameters. Note that by choosing a fraction $f = 0.3$, we are not choosing the best performing network (which would be at $f = 0.9$), but the best generalizing network.

Inputs	POD	FAR	CSI	HSS
MDA	0.44	0.53	0.29	0.41
MDA+NSE	0.47	0.49	0.32	0.45

Table 4: Comparison of the performance of network trained using just the MDA parameters vs a network trained on both MDA+NSE parameters. The networks were trained on 56 cases and tested on an independent set of 27.

Inputs	POD	FAR	CSI	HSS
MDA	0.42	0.51	0.29	0.40
MDA+NSE	0.44	0.52	0.30	0.41

Table 5: Comparison of the performance of network trained using just the MDA parameters vs a network trained on both MDA+NSE parameters. The networks were trained on all 83 cases available. The skill scores reported are on the best validation set.

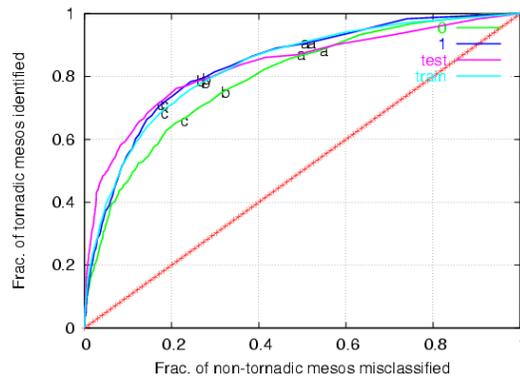


Figure 6: ROC curve showing the performance of a network trained with 56 cases and tested on an independent set of 27 using 30% of the best MDA and NSE features.

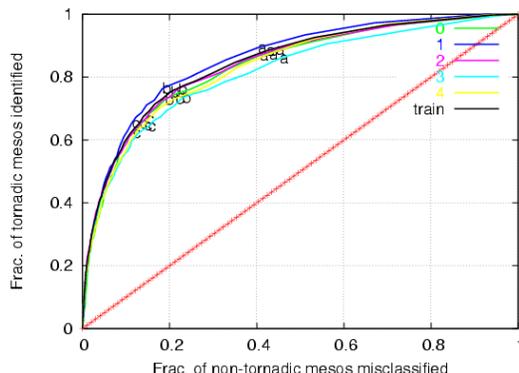


Figure 7: ROC curve showing the performance on various bootstrapped sets of a network trained with all 83 cases using 30% of the best MDA and NSE features.

4. Conclusions

4a. How general are the networks developed in this paper?

As all the preceding ROC plots show, the neural networks developed in this paper (both for MDA and MDA+NSE inputs) achieve similar Heidke skill scores on the training, validation and independent data sets. In fact, in the case of MDA+NSE, we sacrificed higher performance on the training and validation sets in order to be confident about the generalization capability of the networks. The low variability of the ROC plots in this paper also suggest that the neural networks developed in this paper are robust and not over-trained.

4b. Is NSE information helpful?

As discussed in Section 3, the incorporation of NSE parameters into the training process changed the statistics of the data set. While the dataset provided little overfitting in the case of just the MDA parameters, the neural network training regimen would have led to over-fitting if all the NSE parameters were used. Our choice of NSE parameters was driven more by practical constraints (the need to have an operational neural network quickly) than by a need to choose the optimal set of features.

Our answer to the question of whether NSE information is useful is therefore very conservative. We believe (see Table 3 where HSS of 0.5 and CSI of 0.38 are reached) that the incorporation of NSE parameters has the potential to dramatically improve the ability of a neural network to classify which circulations are tornadic. However, in its current state, the neural network that we have developed does not fully realize this potential. The MDA+NSE neural network developed in this paper is only marginally better (see Table 4) than a neural network based purely on radar parameters.

To confirm our diagnosis, we computed the skill score on our independent 27-case test set using the two neural networks, one trained with just 30% of input features, and the other trained with 90% of them. Recall, from Table 3, that the 90% set produced high validation skill scores, and from Figure 5 that the 30% yielded the most general performance. The skill scores on the independent test set using the two f parameters is shown in Table 6. Interestingly, choosing $f = 0.3$ gave us a list of 18 features, very close to the final number of MDA parameters (16).

In this study, we reduced the number of features simply by choosing the best univariate predictors. These are not necessarily the best features to use as inputs. In particular, there could be two NSE features that in combination

Fraction	POD	FAR	CSI	HSS
0.3	0.47	0.49	0.32	0.45
0.9	0.55	0.80	0.17	0.21

Table 6: Comparison of the performance of neural networks trained with different fractions of NSE parameters. The networks were trained on 56 cases and evaluated on an independent set of 27 cases. This analysis merely serves to confirm our choice of a f threshold – it was not used in selecting the threshold.

could be a great predictor, even though each one individually is a poor predictor. Using univariate predictors allowed us to get a handle on this problem, but is not the best possible solution. Also, we chose f empirically by observing where the validation error was closest to the training error. As Figure 5 shows, however, this relationship is not so clear, and this method of choosing the number of features is, at best, an educated guess. Better methods of choosing features as inputs to a NN, by looking at capturing the variability of the data set with as few features as possible, are currently being studied by Varahan et al (**reference here**).

In the neural network (both during training and at run-time), we treat the attributes of each detection independent of all other detections at the same time, and of the same detection at previous times. It might be possible to improve the performance of the neural network if we can optimize the classification of a detection over time. This aspect is being studied by Trafalis et al. (**reference here**).

It is possible that studies which (a) increase the size of the dataset beyond our set of 83 or (b) find better ways to reduce the number of features or (c) use the time history of the detections will find that neural networks that incorporate both radar-based features and NSE inputs will achieve an improvement in tornado classification ability on *independent test data sets*.

5. Acknowledgements

Funding for this research was provided under NOAA-OU Cooperative Agreement NA17RJ1227 and supported by the Radar Operations Center. We gratefully acknowledge the many helpful discussions we have had with Caren Marzban and Don Burgess, both of the University of Oklahoma, on the methods and attributes used in this paper.

References

- Bishop, C., 1995: *Neural Networks for Pattern Recognition*. Oxford.
- Caruana, R., S. Lawrence, and C. Lee Giles: 2001, Overfitting in neural networks: Backpropagation, conjugate gradient, and early stopping. *Neural Information Processing Systems (Proceedings of NIPS*2000)*, MIT Press, volume 13.
- Donaldson, R., R. Dyer, and M. Krauss: 1975, An objective evaluator of techniques for predicting severe weather events. *Preprints, Ninth Conf. on Severe Local Storms*, Amer. Meteor. Soc., Norman, OK, 321–326.
- Heidke, P., 1926: Berechnung des erfolges und der gute der windstarkvorhersagen im sturmwarnungsdienst. *Geogr. Ann.*, **8**, 301–349.
- Krogh, A. and J. Hertz: 1992, A simple weight decay can improve generalization. *Advances In Neural Information Processing Systems*, S. H. Moody, J. and R. Lippman, eds., Morgan Kaufmann, volume 4, 950–957.

- Lakshmanan, V., K. Hondl, G. Stumpf, and T. Smith: 2003, Quality control of weather radar data using texture features and a neural network. *5th Int'l Conf. on Adv. in Patt. Recogn.*, IEEE, Kolkota.
- Marzban, C., 2000: A neural network for tornado diagnosis. *Neural Computing and Applications*, **9**, 133–141.
- Marzban, C. and G. Stumpf, 1996: A neural network for tornado prediction. *J. App. Meteo.*, **35**, 617.
- Masters, T., 1993: *Practical Neural Network Recipes in C++*. Morgan Kaufmann, San Diego.
- Riedmiller, M. and H. Braun: 1993, A direct adaptive method for faster backpropagation learning: The RPROP algorithm. *Proc. IEEE Conf. on Neural Networks*.
- Stumpf, G. and C. Marzban, 2002: Tornado warning guidance. Technical report, Warning Decision Training Branch, National Weather Service, <http://www.wdtb.noaa.gov/resources/PAPERS/twg02/>.
- Stumpf, G., A. Witt, E. D. Mitchell, P. Spencer, J. Johnson, M. Eilts, K. Thomas, and D. Burgess, 1998: The national severe storms laboratory mesocyclone detection algorithm for the WSR-88D. *Weather and Forecasting*, **13**, 304–326.
- Venkatesan, C., S. Raskar, S. Tambe, B. Kulkarni, and R. Keshavamurty, 1997: Prediction of all india summer monsoon rainfall using error-back-propagation neural networks. *Meteorology and Atmospheric Physics*, **62**, 225–240.

6. Appendix

The following 76 attributes were selected from the total list of 245 parameters available as possibly having some role in tornadogenesis and used in the NN training. The first 18 attributes are computed by the MDA while the others are computed by the NSE algorithm. All the MDA attributes are used by the final MDA neural network. Attributes marked with ✓ are used by the final MDA+NSE neural network.

1. Meso Range km
2. Meso base m
3. ✓ Meso depth m
4. ✓ Meso strength rank
5. Meso low-level diameter m
6. ✓ Meso low-level rotational velocity ms
7. ✓ Meso maximum rotational velocity ms
8. Meso height of maximum rotational velocity m
9. ✓ Meso low-level shear mskm
10. ✓ Meso maximum shear mskm

11. Meso height of maximum shear m
12. Meso low-level gate-to-gate velocity difference ms
13. ✓ Meso maximum gate-to-gate velocity difference ms
14. Meso height of maximum gate-to-gate velocity difference m
15. ✓ Meso age min
16. Meso strength index MSI wghtd by avg density of intgrtd lyr
17. ✓ Meso relative depth
18. ✓ Meso low-level convergence ms
19. actual surface pressure mb
20. ✓ height of the 273 K temperature surface m agl
21. average wind speed over a specified depth ms
22. u-component of estimated storm motion vector north-relative ms
23. v-component of estimated storm motion vector north-relative ms
24. estimated 0-3 km storm relative helicity $m^2 s^{-2}$
25. surface relative humidity percent
26. surface virtual temperature Kelvin
27. downdraft CAPE dCAPE for a parcel 1 km AGL
28. dCAPE for a parcel 3 km AGL
29. dCAPE for the parcel at 0 Celsius
30. ✓ Surface CAPE for a parcel with average characteristics over the lowest X mb eg 100 mb
31. Surface CAPE except for CIN
32. Surface CAPE except for LFC
33. Surface CAPE except for EL
34. Surface CAPE except for LI
35. ✓ Surface CAPE except for EHI
36. Surface CAPE except for the average parcel
37. magnitude of the storm-relative flow for the 0-2 km agl layer kts The estimated storm motion vector is subtracted from the average model wind vector of the layer

38. same as above, except for the 4-6 km agl layer
39. same as above, except for the 9-11 km agl layer
40. Bulk Richardson Number
41. BRN shear kts^2
42. Surface CAPE except for the most unstable parcel
43. temperature difference C between 700 and 500 mb
44. magnitude kts of the low-level shear vector surface to 3 km agl
45. average wind speed in the 500-300 mb layer knots
46. maximum theta-e Kelvin in the lowest 300 mb
47. mean lapse rate in the 850-500 mb layer Ckm
48. mean shear
49. deep-layer shear vector magnitude knots
50. average parcel LCL m agl See NOTE above
51. average RH percent below the average parcels LCL
52. ✓ Vorticity Generation Potential VGP using the surface-based CAPE
53. Average Mixing Ratio in 0-1 km layer gkg
54. Average Mixing Ratio in 0-3 km layer gkg
55. Average Mixing Ratio in 0-6 km layer gkg
56. estimated 0-1 km storm relative helicity m^2s^2
57. estimated 0-2 km storm relative helicity m^2s^2
58. Average Relative Humidity in 0-1 km layer
59. 0-1 km shear magnitude knots
60. 0-3 km shear magnitude knots
61. 27 most-unstable parcel
62. 55 most-unstable parcel
63. ✓ 0-18 most-unstable parcel
64. 36-55 most-unstable parcel
65. 82-100 most-unstable parcel

- 66. ✓ Level of Maximum Buoyancy from most-unstable parcel m AGL
- 67. Maximum Buoyancy from most-unstable parcel m^2s^2
- 68. Level of Maximum Buoyancy from most-unstable parcel corresp to 0-6 km shear magnitude knots
- 69. 20 below Level of Maximum Buoyancy from most-unstable parcel corresp to 4-6 km storm-relative flow knots
- 70. 0-6km Bulk Hodograph Curvature
- 71. Low-Level Most-unstable-parcel LFC to Most-unstable-parcel LFC+1km Lapse Rate Ckm
- 72. Normalized most-unstable parcel CAPE divide by $z_{\text{Most-unstable parcel EL}} - z_{\text{Most-unstable parcel LFC}}$
- 73. Most-unstable parcel CAPE Most-unstable parcel CAPE from Most-unstable parcel LFC to Most-unstable parcel LFC+3km
- 74. Most-unstable parcel CAPE from Most-unstable parcel LFC to Most-unstable parcel LFC+3km divided by Most-unstable parcel CAPE age of total
- 75. ✓ Most-unstable parcel CAPE from sfc to 3 km AGL
- 76. ✓ Most-unstable parcel CAPE from sfc to 3 km AGL divided by Most-unstable parcel CAPE age of total