A Neural Network for Damaging Wind Prediction

Caren Marzban[1,2,3]

and

Gregory J. Stumpf[1,2]

[1] National Severe Storms Laboratory, Norman, OK 73069

[2] Cooperative Institute for Mesoscale and Meteorological Studies, University of Oklahoma, Norman, OK 73019

[3] Department of Physics, University of Oklahoma, Norman, OK 73019

Abstract

A neural network is developed to diagnose which circulations detected by the National Severe Storms Laboratory's (NSSL) Mesocyclone Detection Algorithm (MDA) yield damaging wind. In particular, 23 variables characterizing the circulations are selected to be used as the input nodes of a feed-forward, supervised neural network. The outputs of the network represent the existence/nonexistence of damaging wind, based on ground observations. A set of fourteen scalar, non-probabilistic measures, and a set of two multi-dimensional, probabilistic measures are employed to assess the performance of the network. The former set includes measures of accuracy, association, discrimination, skill, and the latter consists of reliability and refinement diagrams. Two classification schemes are also examined.

It is found that a neural network with 2 hidden nodes outperforms a neural network with no hidden nodes when performance is gauged with any of the fourteen scalar measures, except for a measure of discrimination where the results are opposite. The two classification schemes perform comparably to one another. As for the performance of the network in terms of reliability diagrams, it is shown that the process by which the outputs are converted to probabilities allows for the forecasts to be completely reliable. Refinement diagrams complete the representation of the calibration-refinement factorization of the joint distribution of forecasts and observations.

1 Introduction

The recent advances in Neural Network (NN) methodology for modeling nonlinear, dynamical phenomena (Bishop, 1996; Hertz, et al., 1991; Masters, 1993; Müller and Reinhardt, 1991; Ripley, 1996; Warner and Misra, 1996), along with the impressive successes in a wide range of applications, behoove us to investigate the application of NNs for the prediction of atmospheric phenomena. A NN for the prediction of tornados was described in Marzban and Stumpf (1996) and Marzban, et al. (1997). The network was trained to learn the underlying function between 23 attributes, derived from Doppler radar velocity data, and the existence/nonexistence of tornados based on ground observation. It is possible to employ the same attributes to train a NN to predict damaging wind (defined as the existence of either or both of the following conditions: tornado, wind gust $\geq 25 ms^{-1}$.)

In Marzban and Stumpf (1996) it is shown that the NN outperforms several other tornado prediction schemes when the measure of performance is the Critical Success Index, while the Heidke Skill Statistic is employed to compare the performance of the NN with gaussian discriminant analysis in Marzban, et al. (1997). The goal of the present article is to develop a NN for the prediction of damaging wind, and to assess its performance in terms of 14 scalar measures (Marzban, 1997a) as well as 2 multidimensional measures. Since most of the measures considered here are one-dimensional (scalar) quantities, in contrast to the inherently multidimensional nature of forecast quality (Murphy, 1991, 1993, 1996), any attempt to gauge the performance of an NN with these measures is apt to be incomplete. However, in spite of such inherent limitations, many decisions regarding the possible implementation of an algorithm or the winner of a forecasting contest are frequently based on a single one-dimensional measure. For this reason, although the performance of the NN is given in terms of a 2×2 table (with 2 independent degrees of freedom) and in terms of prob-

abilities, this multidimensionality is reduced to a scalar measure. To overcome some of the limitations of scalar, non-probabilistic measures, the multidimensionality is then resurrected through reliability and refinement diagrams (Murphy and Winkler, 1987, 1992; Wilks, 1995).

2 The Mesocyclone Detection Algorithm (MDA)

Algorithms have been designed to detect a variety of severe weather signatures, such as hail, high winds, and tornados in Doppler weather radar data. Mesocyclone detection algorithms are designed to detect the storm-scale circulations which are associated with a region of rotation in thunderstorms; rotating thunderstorms are commonly often associated with tornado occurrence. A mesocyclone detection algorithm resides on the National Weather Service's (NWS) Weather Surveillance Radar - 1988 Doppler (WSR-88D) system, and is used operationally as guidance to meteorologists to warn the general public of tornados and other damaging events associated with supercell thunderstorms.

The National Severe Storms Laboratory (NSSL) has been developing an enhanced Mesocyclone Detection Algorithm (MDA)[1] which contains a variety of new techniques for searching out the patterns within Doppler-radar velocity data which are associated with storm-scale circulations. Previous detection methods were constrained in that particular thresholds and rule-bases were designed to detect only certain types and scales of circulations. Circulations were "thresholded" for dimension (such as depth and height of the base above ground), and for strength (such as rotational velocity). The NSSL MDA relaxes those constraints and is designed to detect a wider spectrum of circulations of varying dimensions and strengths. The main advantage of the new algorithm is two-fold: First, more rotation signatures are

[1]The details of the inner-workings of the MDA are presented elsewhere (Stumpf, et al., 1995).

detected and signatures are more accurately defined. Second, with the detection of additional circulations which may not meet specified rules, statistical methods can be applied to determine the probability that *any* of the detected circulations are associated with damaging wind at the ground.

3 Neural Networks - A Review

There exist many statistical methods for performing classification and regression. Some well-know examples are regression (Draper and Smith, 1981), discriminant analysis (McLachlan, 1992), classification and regression trees (Burrows, 1991), and generalized additive models (Vislocky and Fritsch, 1995). All of these methods are related to NNs in one way or another (Bishop, 1996). There is, however, one feature that sets NNs apart from many other statistical techniques, and that is the way in which they deal with the "curse of dimensionality" (Bishop, 1996): On one hand, a model with a large number of free parameters is desirable because it can approximate a function to any desired accuracy. On the other hand, due to this flexibility, such a model can easily overfit the data and consequently have poor generalization/predictive capability. NNs have the first property in that by increasing the number of hidden nodes (see below), one effectively parametrizes the space of "all" functions (Hornik, et al., 1989). A sufficiently high-order polynomial also has the same feature. However, the advantage of NNs is in the way they control overfitting; the number of free parameters in an M-th order polynomial in n variables grows as n^M, whereas the same number for NNs grows as n^1. Consequently, an NN can learn any function, while maintaining its generalization/predictive capabilities, at least in theory.

Specifically, a feed-forward NN is nothing but a function that maps a set of n variables $x_i, (i = 1, n)$, called input nodes, into a set of variables σ_k, called output nodes. For a NN

with 1 hidden layer, the function itself is parametrized as

$$\sigma_k = f\left(\sum_{i=1}^{H} \omega'_{ik}\ f(\sum_{j=1}^{n} \omega_{ij}\ x_j - \theta_j) - \theta'_k\right),$$

where $\omega_{ij}, \omega'_{ij}, \theta_i$, and θ'_i are all parameters (weights) to be determined from a data set, called the training set. H is called the number of hidden nodes on the hidden layer. The function f is called the activation function, and in the present application it is taken to be the logistic (or fermi) function,

$$f(y) = \frac{1}{1 + \exp(-y)}.$$

The choice of a logistic activation function does not comprise an assumption regarding the underlying function that the NN represents. It can be shown that the performance of an NN is insensitive to the choice of the activation function (Hornik, et al., 1989). This particular choice allows one to relate the NN to logistic regression, since an NN with no hidden layers and with a logistic activation function is equivalent to logistic regression (Bishop, 1996; Masters, 1993; Ripley, 1996).

In principle, for a proper NN development, one requires three independent data sets: a training set, a validation set, and a test set. The validation set may actually be used during the training phase, in order to monitor the predictive performance of the NN, but the test set is to be kept completely out of the training phase. As in all regression methods, the performance of a NN on the training set itself is optimistically biased. The bias can be reduced by evaluating the NN on the validation set, and even further reduced by testing the NN on the test set. However, the price one pays in this process is in the smaller sample size (per set) and increasing variance. In this project, no test set was considered because of the resulting small sample sizes for each of the three data sets and large variances (errors); the estimates of the performance were based on the validation set, and hence, are somewhat

optimistically biased (more so with NNs). The issue of bias versus variance in NNs is examined in (Geman, et al., 1992; Bishop, 1996; Ripley, 1996).

Training an NN involves finding the parameters that minimize some error function; for further details, see Appendix A. Often, and in this application, the mean-square error is chosen as the error function. This choice is motivated by the well-known fact (Bishop, 1996; Draper and Smith 1981) that if the distributions of the dependent variables are normal (gaussian), then least-square estimates are equal to maximum-likelihood estimates. For classification purposes where the dependent variables are often discrete, there exists another error function, called Cross-entropy, that is more natural in that the outputs of a NN trained to minimize Cross-entropy can be arranged to represent the posterior probability of belonging to a class, given the inputs (Bishop, 1996). In the present article, the mean-square error is taken as the error function and posterior probabilities are obtained by estimating the likelihoods of the outputs and by using Bayes' theorem; this method is outlined in Masters (1993) and briefly in Section 5. Other error functions are presently under consideration.

4 Measures of Performance Quality

The scalar, nonprobabilistic measures are derived from the contingency table (otherwise known as the confusion matrix), or in short the C-table,

$$\text{C-table} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} = \begin{pmatrix} \#\text{ of 0's predicted as 0} & \#\text{ of 0's predicted as 1} \\ \#\text{ of 1's predicted as 0} & \#\text{ of 1's predicted as 1} \end{pmatrix},$$
$$= \begin{pmatrix} . & \text{false alarms} \\ \text{misses} & \text{hits} \end{pmatrix}.$$

The total number of nonevents is given by $N_0 = a + b$, that of events is $N_1 = c + d$, and the total sample size is $N = N_0 + N_1$. Note that this table has only 2 degrees of freedom; a general 2×2 matrix has 4 degrees of freedom, but with the 2 constraints $N_0 = a + b$ and

$N_1 = c + d$, that number is reduced to 2. Two common quantities, Probability of Detection (POD) and False Alarm Ratio (FAR), are easily calculated as

$$\text{POD} = \frac{d}{c+d}, \quad \text{FAR} = \frac{b}{b+d}.$$

Specifically, the measures employed in the present analysis are

1. Product of POD and (1-FAR)

$$\text{PRD} = \text{POD} \times (1 - \text{FAR}) = \frac{d}{b + c + d + bc/d} \ ,$$

2. Average of POD and (1-FAR)

$$\text{AVG} = [\text{POD} + (1 - \text{FAR})]/2 = \frac{d}{2}(\frac{1}{c+d} + \frac{1}{b+d}) \ ,$$

3. Fraction Correct

$$\text{FRC} = \frac{a+d}{a+b+c+d} = \frac{a+d}{N} \ ,$$

4. Efficiency

$$\text{EFF} = \frac{a}{a+b} \times \frac{d}{c+d} = \frac{a}{N_0} \times \frac{d}{N_1} \ ,$$

5. Critical Success Index

$$\text{CSI} = \frac{d}{b+c+d} \ ,$$

6. True Skill Score

$$\text{TSS} = \frac{ad - bc}{(a+b)(c+d)} = \frac{det\ C}{N_0 N_1} \ ,$$

7. Heidke's Skill Score

$$\text{HSS} = \frac{2(ad - bc)}{(a+b)(b+d) + (a+c)(c+d)} = \frac{2\ det\ C}{N_0(b+d) + N_1(a+c)} \ ,$$

8. Gilbert's Skill Score

$$\text{GSS} = \frac{ad - bc}{(ad - bc) + (a + b + c + d)(b + c)} = \frac{det\ C}{det\ C + N(b + c)} \ ,$$

9. Clayton's Skill Score

$$\text{CSS} = \frac{ad - bc}{(a + c)(b + d)} = \frac{det\ C}{(a + c)(b + d)} \ ,$$

10. Doolittle's Skill Score

$$\text{DSS} = \frac{(ad - bc)^2}{(a + b)(a + c)(d + b)(d + c)} = \frac{(det\ C)^2}{N_0 N_1 (a + c)(b + d)} \ ,$$

11. Discrimination Measure

$$\text{DIS} = (\frac{N_0}{N})^2 \left[1 + \frac{2d}{N}(1 + \frac{d}{b})\right] + (\frac{N_1}{N})^2 \left[1 + \frac{2a}{N}(1 + \frac{a}{c})\right] ,$$

$$\text{DIS} = (\frac{N_0}{N})^2 \left[1 + \frac{2c}{N}(1 + \frac{c}{a})\right] + (\frac{N_1}{N})^2 \left[1 + \frac{2b}{N}(1 + \frac{b}{d})\right] ,$$

for $ad - bc \geq 0$, and $ad - bc < 0$, respectively. We also define three new measures, a quantity called δ, and a pair of angles θ and ϕ

12.

$$\delta = |\frac{d}{c + d} + \frac{b}{b + d} - 1| \ ,$$

13.

$$\theta = \frac{1}{2} tan^{-1} \frac{2(ab + cd)}{d^2 + b^2 - a^2 - c^2} \ ,$$

14.

$$\phi = \frac{1}{2} tan^{-1} \frac{2(ac + bd)}{d^2 + c^2 - a^2 - b^2} \ .$$

The following multidimensional measures complete the list of measures considered:

15. Reliability Diagram, and

16. Refinement Diagram .

Unlike the other measures, the last three scalar measures are "measures of error" in that lower values correspond to better performance. Although they too can be transformed into "measures of success" that would obfuscate their geometrical interpretation (shown below). These scalar measures have been examined in Marzban (1997a). The last two measures appear in the calibration-refinement factorization (Murphy and Winkler, 1987) of the joint probability $p(f, \sigma)$. Also see Wilks (1995).

Given the multidimensionality of forecast quality, it is unwarranted to restrict any analysis to a single measure of forecast quality, and as a result all of the measures will be computed. Reliability and refinement diagrams will be computed as well.

Clearly, all such measures can be calculated from the outputs of an NN directly, but the output nodes of the NN described thus far do not necessarily have a probabilistic interpretation - a desirable feature. The next section outlines the transformation of the NN outputs into probabilities.

5 Probabilities

Given the outputs of a NN, it is possible to derive Probability Density Functions (PDFs) for the NN outputs, and from them one can construct event probabilities in a Bayesian formalism. As discussed in the next section, the NN employed for the present analysis has two output nodes. First, the two output nodes, σ_{left} and σ_{right}, are combined into a single fictitious output node, σ:

$$\sigma = f(\beta(\sigma_{left} - \sigma_{right})), \tag{1}$$

where f is the logistic function, and β is a parameter that measures the strength of the mixing; we have found that the ultimate results are quite insensitive to the exact value of the β parameter. Then, estimates of the PDFs are arrived at by a method proposed by Parzen (1962). There, a PDF estimator is shown to be

$$L(\sigma) = \frac{1}{n\lambda} \sum_i W(\frac{\sigma - \sigma_i}{\lambda}),$$

where σ_i are a random sample of size n, and the W is a weighting function. Whereas for a broad range of Ws, Parzen's estimator asymptotically approaches the true density function as the sample size increases, a common choice is the gaussian function

$$W = \exp^{-[(\sigma - \sigma_i)/\lambda]^2},$$

which we adopt. Note that this choice does not imply that the PDFs themselves are gaussian. Here, the random sample σ_i is drawn from the training data. In other words, n is the size of the training set, and the σ_i are the values of the single fictitious output node that result from exposing the trained NN to the training data. The parameter λ is a "smoothing parameter" that is to be fixed; the final results are insensitive to the specific value of λ, as well. Figure 1a shows an example of the distribution of the network's fictitious output (at $\beta = 1.0$) for the "0"'s and the "1"'s, and Figure 1b shows the corresponding Parzen's estimator at $\lambda = 0.1$. These likelihood functions can be employed to obtain *posterior* event probabilities, $P_1(\sigma)$, via Bayes' theorem

$$P_1(\sigma) = \frac{p_1 L_1(\sigma)}{p_0 L_0(\sigma) + p_1 L_1(\sigma)},$$

where $p_0 (= 1 - p_1), p_1$ are the *prior* probabilities for nonevents and events, respectively. $P_1(\sigma)$ is the conditional probability of an event, given the value of the output, σ; it is what Murphy, et al. (1989) call "the forecast, f". Mathematically, $P_1(\sigma) = p(\text{event}|\sigma) = f$.

In addition to using the output nodes directly for performing classification, these probabilistic forecasts can again be reduced to dichotomous values allowing for an alternative computation of C-tables. Consequently, all 14 scalar measures can also be re-calculated also from these "smoothed" C-tables. This will be explained in the next section.

6 The Method

Any circulation detected on a particular volume scan of radar data (the sampling rate of a radar volume scan is approximately 6 minutes) can be associated with a report of a tornado and/or winds in excess of $25ms^{-1}$ (i.e., damaging wind). If a circulation is detected within 20 minutes prior to a ground report of damaging wind or within 5 minutes after a report, the circulation is classified as a "prediction" of damaging wind. The motivation for treating the circulations in the 5-minute interval after the ground report as "damaging wind" is the possibility of small errors in the reporting times of the actual events. Therefore, the neural network is trained to provide up to a 20-minute "lead-time" for damaging wind warnings by the NWS.

A list of the 23 input variables, along with a brief description of each quantity is provided in Appendix B. The values for these quantities were linearly scaled to lie in the range 0.1 to 0.9 and were then presented to the NN as inputs. This scaling is done to equalize the a priori contribution of all the variables, and these limits were chosen (instead of 0 and 1) to avoid numerical singularities. The computer codes employed for damaging wind prediction are the same as those developed for tornado prediction.

Two classification schemes are possible: We shall refer to them as the "discrete", and the "smoothed" method, respectively. In the former, a C-table can be obtained directly from the 2 output nodes of the network in what is referred to as the "winner-takes-all" method

(Bishop, 1996). In this method, there are as many output nodes as the number of classes (i.e. 2), and the output with the higher activation designates the class. This discrete method of classification is quite common and leads to one type of C-table.

As mentioned above, in this method a quantity that must be determined is the relative class size (i.e., the ratio of the number of events to non-events) in the training set. In this study, its optimal value was obtained by training a variety of networks with training sets of different class sizes, and selecting the one with the highest *validation* performance.

A second type of C-table can be obtained by considering the conditional probability of a given class, given the outputs. In this method, the posterior probabilities computed in the previous section can be employed to form smoothed estimates that can then be reduced to a dichotomous one, i.e. a C-table. This method has been shown to reduce variance (Glick, 1978). The reduction to the dichotomous case is performed not by imposing and varying an arbitrary threshold on the output nodes, but by fixing the posterior probability threshold at 0.5 (50%) and instead varying the parameter p_1 (Marzban, 1997b).

The reduction is performed only to allow for the computation of the scalar measures. In a 2-class problem, as long as the costs of misclassification are assumed equal (as they are in this article) the only probability threshold that makes any sense is at a posterior probability of 50%, because $P_0(\sigma) < P_1(\sigma)$ or $P_0(\sigma) > P_1(\sigma)$ decides the group to which σ belongs, where $P_0 + P_1 = 1$. It is important to emphasize that after the reduction, p_1 effectively plays the role of a threshold (Marzban, 1997b), and as such it is *not* equal to the climatological probability as given by N_1/N. In the reduced posterior probabilities, p_1 is simply a parameter of the model that must be determined in some fashion.[2] One way in which a value of p_1 may be

[2]There is one other reason why p_1 can be treated as a parameter: The performance of all regression models (including NNs) on the validation or the test set depends on N_1/N of the training set. However,

selected is by picking the one that optimizes some measure of performance. In this method, since the performance measures are based on reduced posterior probabilities, there is no need to withhold any data from the NN during training by controlling the class sizes - the more data, the better the estimates of the probability distributions. Instead, the parameter that is varied is p_1, and the object of this method is to find the optimal value of this parameter.

The advantage of the first ("discrete") method is its simplicity and robustness in that no distributions must be estimated, and the advantage of the second ("smoothed") method is in utilizing all the available data for training the NN.

From these two types of C-tables one can calculate all of the 14 scalar measures. As a result, two sets of scalar measures will be computed - one from each type of C-table; the measures computed from the "discrete" C-tables will be labeled as PRD, AVG, etc., and those based on the "smoothed" C-tables will be distinguished by a prime (e.g. PRD', AVG', etc.).

Another quantity that must be determined in both methods is the number of hidden nodes. Here, it is found by testing NNs with a variety of number of hidden nodes, and selecting the one that yields the highest performance when the network is exposed to the *validation* set. This will preclude any overfitting of the training set. It may be objected that this method of finding the optimal number of hidden nodes, though not overfitting the training set, may overfit the validation set instead. However, as discussed below, this outcome is precluded since 20 randomly selected validation sets (and training sets) were examined (see "bootstrapping" or "cross-validation" in Bishop 1996, p. 372-375).

---

if the output node of a neural network is not required to have a probabilistic interpretation, there is no reason why the optimal value of N_1/N should be equal to the true prior probability, i.e. the climatological probability (Bishop, 1996).

The training and the validation sets were selected from a total of 1946 circulations detected by the MDA. The number of damaging wind circulations (i.e., events, or "1"s) was 368, and the remaining 1578 circulations were nondamaging wind, making for a climatological ratio of 368/1946 = 0.189. The 368 "1"s were divided into two groups of 246 and 122 cases to be used in the training set and the validation set, respectively. The same climatological ratio, 0.189, was employed to select 524 "0"s in the validation set.

In the first method, the number of nondamaging cases in the training set, N_0^T, was varied from 100 to 1,054 (= 1578 - 524), in increments of 200, and the NN was tested on the respective validation set each time. . In short,

$$\begin{pmatrix} \text{no. of 0's in training} = N_0^T & \text{no. of 0's in validation} \\ \text{no. of 1's in training} & \text{no. of 1's in validation} \end{pmatrix} = \begin{pmatrix} 100 < N_0^T < 1054 & 524 \\ 246 & 122 \end{pmatrix}.$$

In the second approach, since the classification criterion is based on the distributions of σ (equation 1), and on the posterior probabilities derived therefrom, there is no reason to withhold any data from the NN during training. As a result, all 1,054 nonevents and 246 events were used for training. Note that the climatological class ratio of 0.189 was used both in the training and validation sets. Then p_1 was varied from 0.1 to 0.9 in 0.1 increments, and the validation measures were calculated. In this approach, since the forecasts are probabilistic, reliability and refinement diagrams were also computed.

In order to assure that the selection of the circulations for either training or validation was not biased, and to preclude overfitting the validation sets, the entire procedure was repeated for 20 different random sets (training and validation). The validation measures were then averaged over the different random sets. It is important to note that both the training set and the validation set were randomly selected, and so each of the 20 attempts represents an independent sampling of the data.

7 Results and Conclusions

The graphs in Figure 2 show the NN results. The y-axis of each plot is the value of a measure, and they are computed by averaging the measures for the 20 different partitions of the validation data sets. Each plot has 3 curves corresponding to 0, 2, and 4 hidden nodes on one hidden layer, and the error-bars on each curve display the 90% confidence interval. The "un-primed" measures are computed from the "winner-takes-all" C-tables, and so are plotted as a function of the non-event sample size in the training set, N_0^T. The "primed" measures are computed from the dichotomized probabilistic forecasts, and so are plotted as a function of the parameter p_1.

Based on the scalar measures, evidently, the NN with 2 hidden nodes reaches higher (or equal) performance values than the NN with 0 or 4 hidden nodes. This is true for all the scalar measures of performance, except DIS (and DIS'), in terms of which the NN with 0 hidden nodes reaches higher performance.

As for N_0^T and p_1, the corresponding optimal values depend on the particular measure of performance. This is not surprising, given that the various scalar measures gauge different aspects of performance.

In Figure 2, by comparing the measures as obtained in the two classification schemes (i.e., the discrete and the smoothed), it can be seen that the two classification methods are equivalent in that the best performance as a function of the number of nonevents in the training set is matched by a similar performance as a function of p_1. In other words, the optimal NN in the first method performs comparably to that in the second method, in spite of the larger training set employed in the latter.

Figures 3a and 3b are the reliability diagram and the refinement diagram for an NN with 2 hidden nodes for $p_1 = 0.1, 0.2, ..., 0.9$. For clarity, the diagrams for 0 and 4 hidden

nodes are not plotted, but they are statistically equivalent to those of the NN with 2 hidden nodes. From the reliability diagram it is evident that $p_1 = 0.2$ produces reasonably reliable forecasts. This is expected because $p_1 = 0.2$ is approximately equal to the climatological prior probability of 0.189. In fact, as seen from Figure 4a, showing the 90% confidence intervals for the reliability diagram with p_1 set to its climatological value, these forecasts are completely reliable.

Figure 3b shows that higher values of p_1 yield more "refined" forecasts (Murphy and Winkler, 1987) than those with lower values of p_1 in that the former are more U-shaped. When p_1 is set to its climatological value, the sharp peak at 10% in the refinement plot (Figure 4b) suggests that the forecasts are not very refined; however, it must be noted that this behavior is a simple consequence of the abundance of nonevents in the data set.

It is worth noting that a comparison of Figures 3a and 3b indicates that the most reliable forecasts ($p_1 = 0.189 \sim 0.2$) are in fact not the most refined forecasts ($p_1 = 0.9$). This too is not surprising, because reliability and refinement are two independent quantities.

The skill scores, HSS, GSS, and DSS, and the measures, PRD, and CSI, all behave quite similarly (Figure 2). Of course, the reason may be that they are correlated in that they gauge similar facets of performance. Whatever the reason, they are optimized at the same value of N_0^T or p_1, i.e. $p_1 \sim 0.3$. Interestingly, according to the error-bars, performance at $p_1 = 0.2 (\sim$ climatology) is statistically equivalent to that at $p_1 = 0.3$, and the former is also the value of p_1 that yields complete reliability (Figure 4a).

8 Discussion

Before proceeding, the odd behavior portrayed in CSS', DSS', and DIS' requires an explanation. The extrusion of the curves from the bounds of these figures is meant to reflect the

existence of an upper-bound in p_1, beyond which the measures are undefined! It is easy to understand this phenomenon; as mentioned previously, p_1 represents the value of the decision threshold (i.e., the value of the fictitious output node, σ), that separates the events from the nonevents. Pictorially, this threshold is the crossing point of the curves $p_0 L_0$ and $p_1 L_1$, and not that of L_0 and L_1 (see Figure 1b), because it is the former quantities that are proportional to the posterior probabilities. The upper-bound in p_1 occurs when the curve $p_0 L_0$ is entirely contained under the $p_1 L_1$ curve. In this situation, a decision threshold does not exist (Marzban, 1997).

In order to assure that the 3 curves in each graph of Figure 2 are statistically distinct, and also to obtain a statistical bound on the measures, the 20 outcomes (for a given training set and number of hidden nodes) were used to calculate the 90% confidence intervals. These appear as the error bars on the various curves in Figure 2. It can be seen that throughout the range of the curves the differences between the three curves are statistically significant at the 90% confidence level. The few exceptions where the three curves overlap occur far from the the critical values (where the measures are optimized) and are, therefore, of no concern here. What this implies is that the NN with 2 hidden nodes outperforms the NN with 0 hidden nodes regardless of the measure of performance and the method of classification. The only exception is DIS (and DIS'), in terms of which the situation is reversed. It is not impossible for one algorithm to outperform another algorithm in terms of one scalar measure, and not in terms of another. This is simply due to the one-dimensional nature of such measures as a given scalar measure captures only one facet of performance quality. Finally, the relative position of the three curves corresponding to 0, 2, and 4 hidden nodes indicates that the first is underfitting the data, the last is overfitting it, while the 2 hidden node curve is the optimal fit.

The determination of the optimal values of N_0^T and p_1 is not so straightforward, even though we may now concentrate only on the 2-hidden-node curves. Not surprisingly, as is evident from the graphs, that choice depends on the particular measure. Whereas some measures are optimized when only 300 nonevents are included in the training set, others require 1100 cases. On the other hand, some appear to reach optimum at 700, while others do not exhibit a true optimum at all in the examined range. The same patterns repeat for the measures based on "smoothed" C-tables; whereas some measures are optimized for $p_1 = 0.1$, others optimize at $p_1 = 0.8$, and yet others exhibit a rather flat plateau. That every measure has its "preferred" threshold has been analytically proven in the case of gaussian models (Marzban, 1997b). Also, as mentioned previously, the existence of correlations among some of the measures accounts for the similar values of p_1 or N_0 at which some of the measures are optimized.

Finally, a question arises that also points out a limitation of this study, namely "why examine all the different measures when the NN is trained to minimize only the mean square error?" One answer can be found by noting that the mean square error is an analytic function of ω. This is important because $\partial E/\partial\omega$ is necessary for the operation of any gradient learning rule, such as Conjugate Gradient. The considered measures, however, are discrete and do not lend themselves to gradient methods. A solution may be to deduce analytic expressions for the measures as functions of ω which reduce to the canonical expressions of Section 4 in the binary case, and employ them as error functions during the training phase. Unfortunately, there is some ambiguity in defining such analytic measures because there exist many distinct analytic choices that reduce to a given measure. An example is provided by FRC: note that when the target values and the predicted values are binary (0 or 1), then not only

$$E(\omega) = \frac{1}{N}\sum_{i=0}^{N}[t_i - p_i(\omega)]^2 \rightarrow (1\text{-FRC}),$$

but also

$$E'(\omega) = \frac{1}{N}\sum_{i=0}^{N}[(1-t_i)(1-p_i(\omega)) + t_i p_i(\omega)]^1 \rightarrow \text{(1-FRC)}.$$

Therefore, if one were interested in the validation FRC as a measure of performance, then one could train the NN to minimize two different, but analytic, analogs of FRC, i.e. $E(\omega)$ or $E'(\omega)$.

To recapitulate the findings of the present article, in addition to the neural network devised for tornado prediction described in Marzban and Stumpf (1996) and Marzban, et al. (1997), a similar network is developed for damaging wind prediction. Two methods of classification, fourteen scalar performance measures, and two probabilistic, multidimensional measures are considered. A NN with 2 hidden nodes is found to be optimal for thirteen of the fourteen measures of performance and both methods of classification; the exception is the measure of discrimination, DIS (and DIS'), for which a NN with zero hidden nodes is optimal. It is shown that, not surprisingly, the ultimate choice of the optimal network, as determined either from the ratio of the number of events to nonevents in the training set or the value of the parameter p_1, is contingent on the particular scalar measure. Fortunately, there exists a unique NN that optimizes several of the measures as well as yielding completely reliable forecasts.

Appendix A

In this appendix we present some of the details of the NN training.

The algorithm begins with a random choice of weights. It then performs Simulated Annealing (SA) (below, and Masters, 1993) to find another set of weights that give a lower value of the error function. This is repeated for a number of times (taken to be three, here) to assure that the best possible weights are obtained. In fact, one can use SA to find

the global minimum (or a deep, local minimum) of the error function; however, this is a slow process, and so instead, after a set of weights are selected by SA, Conjugate Gradient goes into action until no improvement is found in the error function. At this point, SA is called on again to search for possibly better weights. If any are found, conjugate gradient is employed to reach the lowest error function; otherwise the algorithm begins with an entirely new random choice of weights and repeats the entire process. This process is designed to both avoid and escape local minima , and it terminates upon the trainer's interrupt signal. In principle, one can never be certain that a global minimum has been found. However, given that the classification problem is inherently statistical (i.e., that the data is noisy) it is not necessary to find the global minimum; a deep, local minimum may be statistically equivalent to a global one. One may even argue that it is not the global minimum of the training error function that is important, but the "minimum" of the training error function *and* the validation error function, simultaneously.

Here, we reproduce a paragraph from (Marzban and Stumpf, 1996) that offers a brief and intuitive interpretation of SA: Imagine a mountainous landscape, consisting of great many local minima. Now imagine that there is a ball resting on this landscape which we would like to place in the deepest of these minima. It is a simple task to prove that the chances of succeeding in this task are maximized if the landscape is shaken, first violently, then less violently, followed by even gentler and gentler shakes. This process is referred to as the annealing process. In the NN context, the initial weights are selected from a random distribution whose width is decreased systematically, analogous to the systematic decrease in the strength of shaking the box in the above example. In this way, one can obtain the best set of initial weights with the hope of avoiding the local minima. Of course, since the proof of SA's success is a probabilistic one, the method does not assure success upon a single

attempt. When the system has landed in a local minimum, one can use annealing again to find a better/deeper minimum. In this way, one expedites finding the global minimum, or at least a sufficiently deep local minimum.

Appendix B

A brief description of the 23 input variables is as follows:

1. Base: The height AGL of the bottom of the circulation.

2. Depth: The depth of the circulation.

3. "Strength Rank": A non-dimensional number related to the range dependent strength parameters (rotational velocity and shear) of a circulation. Each 2D feature used to build a 3D detection has a Strength Rank. The Strength Rank of the 3D detection is the value at which a continuous depth ($> 3\ km$, base $< 5\ km$ AGL) of 2D features used to make up this 3D detection are greater than or equal to this value.

4. Low-altitude diameter: Diameter of the circulation at its lowest elevation detected.

5. Maximum diameter: The largest diameter of the circulation throughout its entire depth.

6. Height of maximum diameter: The height AGL of the largest diameter of the 3D circulation.

7. Low-altitude rotational velocity: Rotational Velocity [(max outbound - max inbound)/2] of the circulation at its lowest elevation detected.

8. Maximum rotational velocity: The largest rotational velocity of the circulation throughout its entire depth.

9. Height of maximum rotational velocity: The height AGL of the largest rotational velocity of the 3D circulation.

10. Low-altitude shear: Shear [(max outbound - max inbound)/diameter] of the circulation

at its lowest elevation detected.

11. Maximum shear: The largest shear of the circulation throughout its entire depth.

12. Height of maximum shear: The height AGL of the largest shear of the 3D circulation.

13. Low-altitude gate-to-gate velocity difference: Gate-to-gate velocity difference (largest velocity difference between adjacent gates at constant range within the 2D feature) of the circulation at its lowest elevation detected.

14. maximum gate-to-gate velocity difference: The largest GTGDV of all 2D features within the circulation detection throughout its entire depth.

15. Height of maximum gate-to-gate velocity difference: The height AGL of the maximum GTGDV of the 3D circulation.

16. Core base: The height of the lowest 2D feature who's Strength Rank is greater than or equal to the Strength Rank (#3, above).

17. Core depth: The continuous depth of the 2D features in the 3D detection whose Strength Rank is greater than or equal to the Strength Rank (#3, above).

18. Age: The age of the circulation.

19. MSI: Vertically Integrated Strength Ranks for all 2D features within the 3D detection.

20. MSI rank: Rank derived using the vertically integrated rotational velocity and shear within the 3D detection.

21. Relative depth: The ratio between the depth of the circulation and the depth of the entire storm cell.

22. Low-altitude convergence: The average radial convergence in the vicinity of the circulation integrated over a depth of $0 - 2$ km AGL.

23. Mid-altitude convergence: The average radial convergence in the vicinity of the circulation integrated over a depth of $2 - 4$ km AGL.

Acknowledgments

C. M. is grateful to H. Brooks, C. Doswell, J. Kuehler, V. Lakshmanan, and A. Murphy for discussions on measures of performance. We also thank Mike Eilts and Arthur Witt for a careful reading of an original version of this manuscript. Partial support was provided by the FAA and the NWS/OSF.

References

Bishop, C. M., 1996: *Neural Networks for Pattern Recognition.* Clarendon Press, Oxford, pp 482.

Burrows, W. R., 1991: Objective guidance for 0-24-hour and 24-48-hour mesoscale forecasts of lake-effect snow using CART. *Wea. Forecasting*, **6**, 357-378.

Draper, N. R. and H. Smith, 1981: *Applied Regression Analysis.* John Wiley and Sons, New York, 709 pp.

Geman, S., E. Biensenstock, and R. Doursat, 1992: Neural networks and the bias/variance dilemma. *Neural Computation*, **4**, 1-58.

Glick, N., 1978: Additive estimators for probabilities of correct classification. *Pattern Recognition*, **10**, 211-222.

Hertz, J., A. Krogh, and R. G. Palmer, 1991: *Introduction to the Theory of Neural Computation.* Addison-Wesley, 414 pp.

Hornik, K., M. Stinchcombe, and H. White, 1989: Multilayer feedforward networks are universal approximators. *Neural Networks*, **4:2**, 251-257.

Marzban, C., 1997a: Scalar measures of performance in rare-event situations. To appear in *Wea. Forecasting.*

Marzban, C., 1997b: Bayesian prior probability and scalar performance measures in gaussian models. To appear in *Journal of Applied Meteorology.*

Marzban, C., H. Paik, and G. Stumpf, 1997: Neural networks vs. gaussian discriminant analysis. *AI applications*, **11**, 49-58.

Marzban, C., and G. Stumpf, 1996: A neural network for tornado prediction based on Doppler radar-derived attributes. *Journal of Applied Meteorology*, **35**, 617-626.

Masters, T., 1993: *Practical Neural Network Recipes in C++.* Academic Press, 493 pp.

McLachlan, G. J., 1992: *Discriminant Analysis and Statistical Pattern Recognition*, John Wiley and Sons, Inc., New York. 526 pp.

Müller, B., and J. Reinhardt, 1991: *Neural Networks: An Introduction.* Springer-Verlag: The Physics of Neural Networks Series, 266 pp.

Murphy, A. H., 1991: Forecast verification: Its complexity and dimensionality. *Mon. Wea. Rev.*, **119**, 1590-1601.

Murphy, A. H., 1993: What is a good forecast? An essay on the nature of goodness in weather forecasting. *Wea. Forecasting*, **8**, 281-293.

Murphy, A. H., 1996: The Finley affair: a signal event in the history of forecast verification. *Wea. Forecasting*, **11**, 3-20.

Murphy, A. H., B. G. Brown, and Y-S. Chen, 1989: Diagnostic verification of temperature forecasts. *Wea. Forecasting*, **4**, 485-501.

Murphy, A. H., and R. L. Winkler, 1992: Diagnostic verification of probability forecasts. *Int. J. Forecasting*, **7**, 435-455.

Murphy, A. H., and R. L. Winkler, 1987: A general framework for forecast verification. *Mon. Wea. Rev.*, **115**, 1330-1338.

Parzen, E., 1962: On estimation of a probability density function and mode. *Ann. Math. Statistics*, **33**, 1065-1076.

Ripley, B. D., 1996: *Pattern Recognition and Neural Networks.* Cambridge: University Press.

Stumpf, G., C. Marzban, and E. N. Rasmussen, 1995: The new NSSL Mesocyclone Detection Algorithm: A paradigm shift in the understanding of storm-scale circulation detection. *27th Conference on Radar Meteorology*, Vail, CO, Amer. Meteor. Soc., in press.

Vislocky, R. L., and J. M. Fritsch, 1995: Generalized additive models versus linear regression in generating probabilistic MOS forecasts of aviation weather parameters. *Wea. Forecasting*, **10**, 669-680.

Warner, B., and M. Misra, 1996: Understanding neural networks as statistical tools. *The American Statistician*, **50**, 284-293.

Wilks, D. S., 1995: *Statistical Methods in the Atmospheric Sciences.* Academic Press, NY. 467 pp.

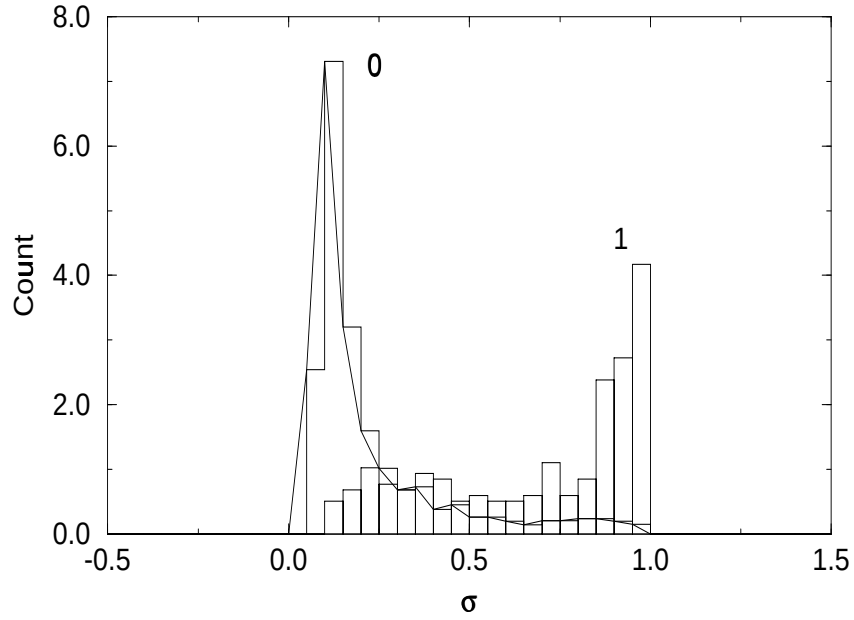
Figure Captions

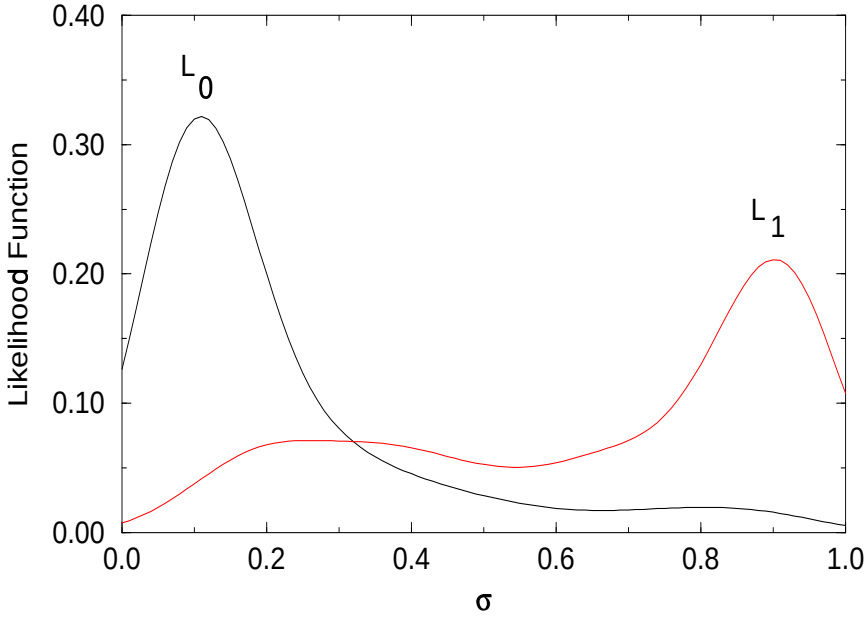
Figure 1. (a) The distribution of the NN's single "fictitious" output node, σ, and (b) Parzen's PDF estimates at $\lambda = 0.1$.

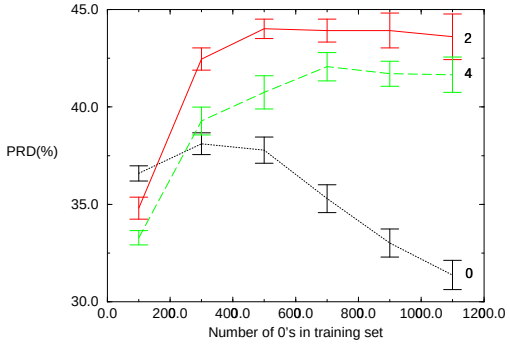
Figure 2. The average validation measures for networks with 0, 2, and 4 hidden nodes on one hidden layer. The unprimed measures are based on a "winner-takes-all" method of classification, and the primed measures are based on dichotomized probabilistic forecasts. Also shown are the 90% confidence intervals based on 20 randomly selected samples. The vertical lines mark the climatological values of N_0^T and p_1.

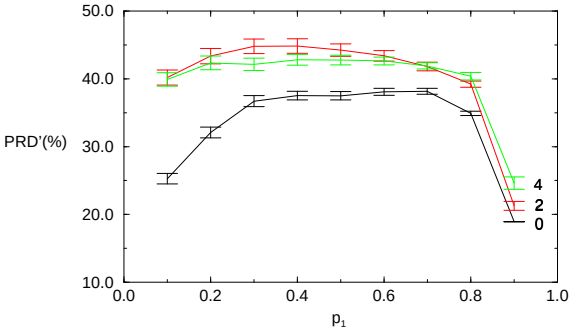
Figure 3. The reliability diagram (a), and the refinement diagram (b) of an NN with 2 hidden nodes, for $p_1 = 0.1, 0.2, 0.3, \ldots, 0.9$. The diagonal line in (a) represents perfect reliability.

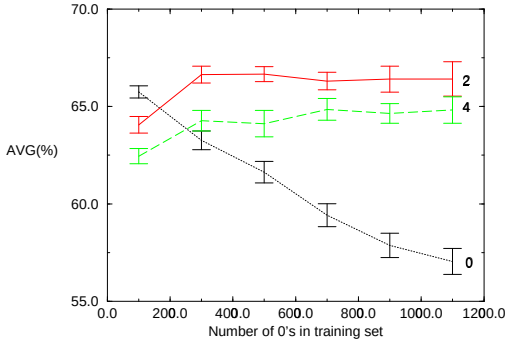
Figure 4. The reliability diagram (a), and the refinement diagram (b) of an NN with 2 hidden nodes, and p_1 given by its climatological value, $p_1 = 0.189$. The error-bars are the 90% confidence intervals based on 20 randomly selected samples.

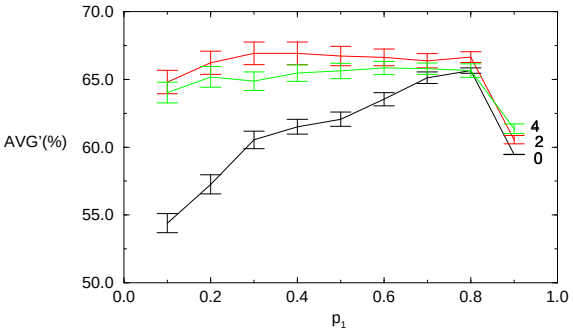

8.0
6.0
4.0
2.0
0.0
Count
0
1
-0.5
0.0
0.5
1.0
1.5
σ

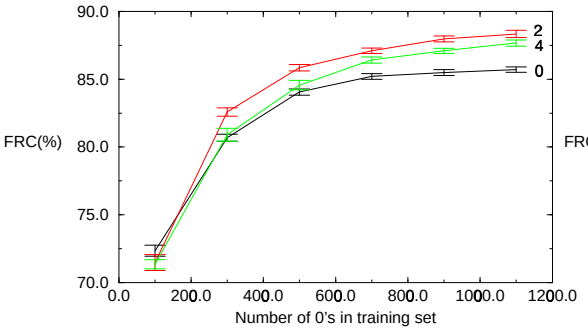

0.40
0.30
0.20
0.10
0.00
Likelihood Function
L 0
L 1
0.0
0.2
0.4
0.6
0.8
1.0
σ

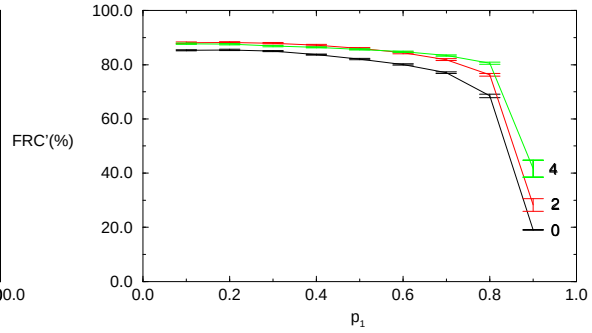

PRD(%)
Number of 0's in training set
2
4
0

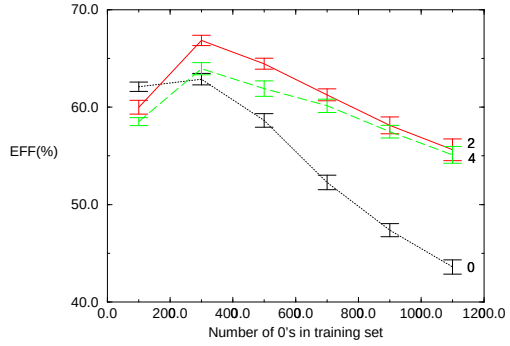

PRD'(%)
p_1
4
2
0

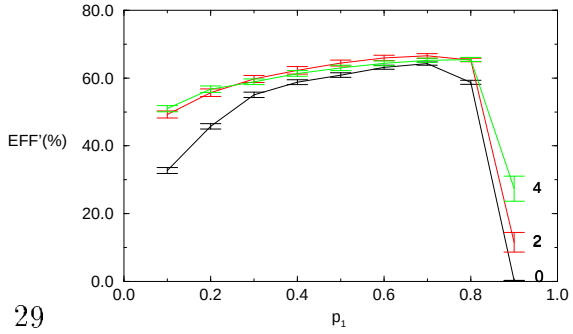

AVG(%)
Number of 0's in training set
2
4
0

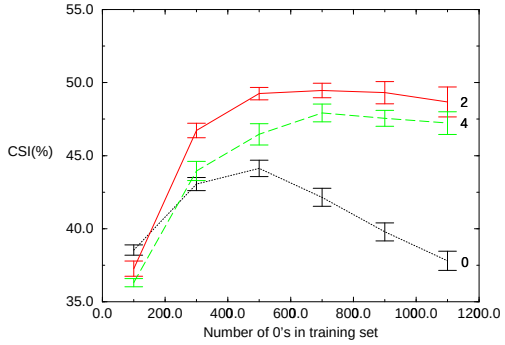

AVG'(%)
p_1
4
2
0

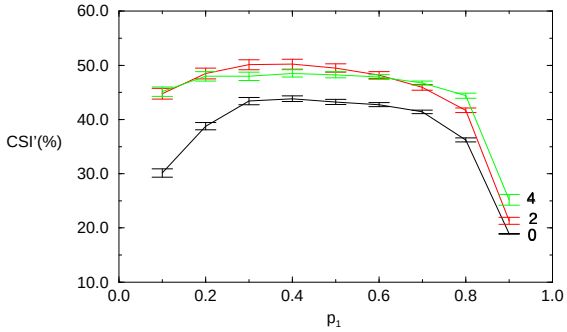

FRC(%)
Number of 0's in training set
2
4
0

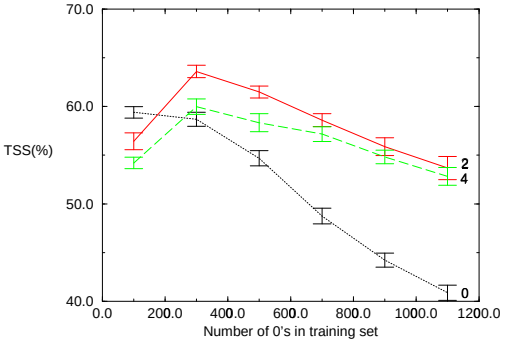

FRC'(%)
p_1
4
2
0

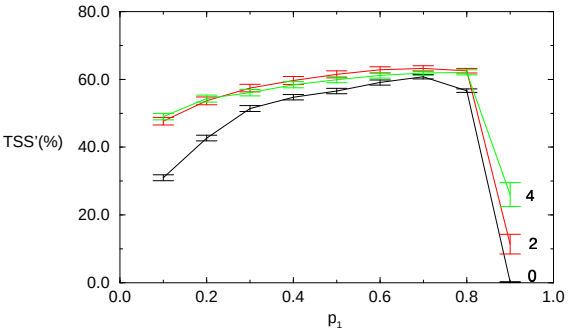

EFF(%)
Number of 0's in training set
2
4
0

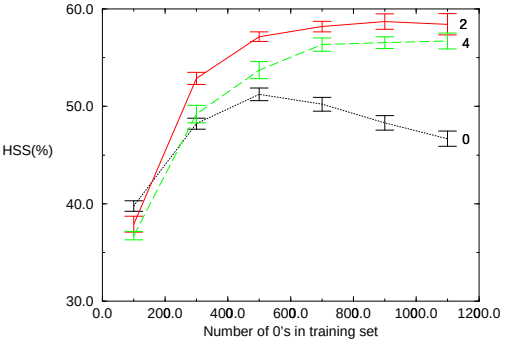

EFF'(%)
p_1
4
2
0

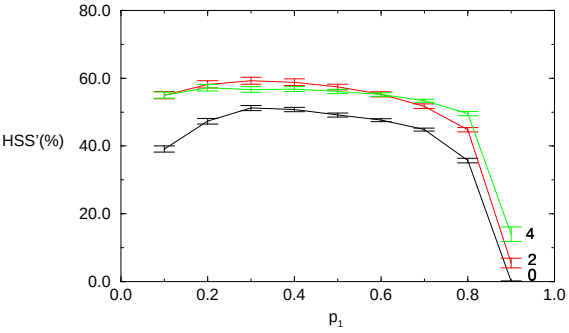

CSI(%)
Number of 0's in training set
2
4
0

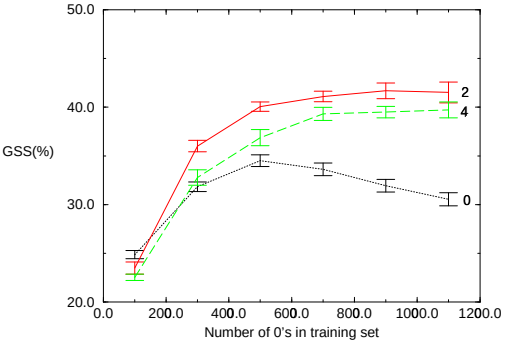

CSI'(%)
p1
4
2
0

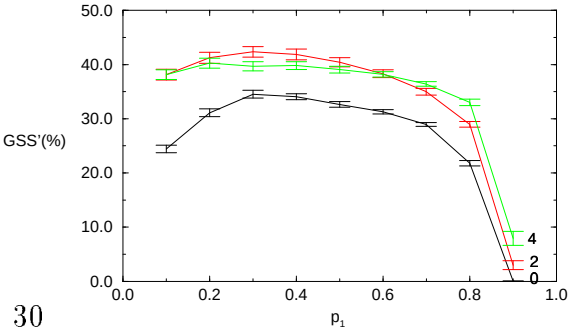

TSS(%)
Number of 0's in training set
2
4
0

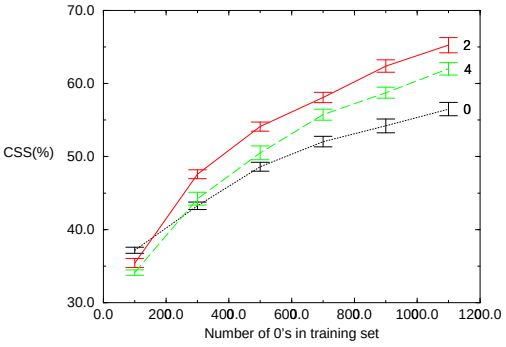

TSS'(%)
p1
4
2
0

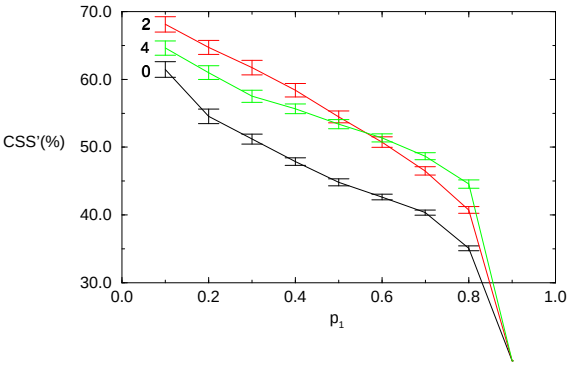

HSS(%)
Number of 0's in training set
2
4
0

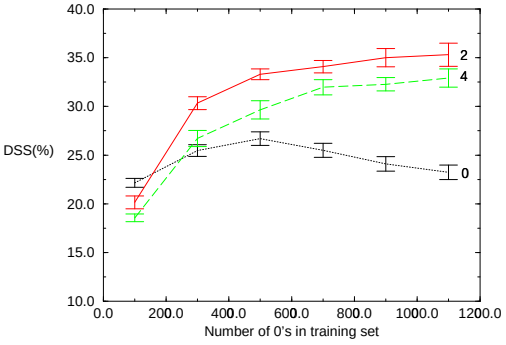

HSS'(%)
p1
4
2
0

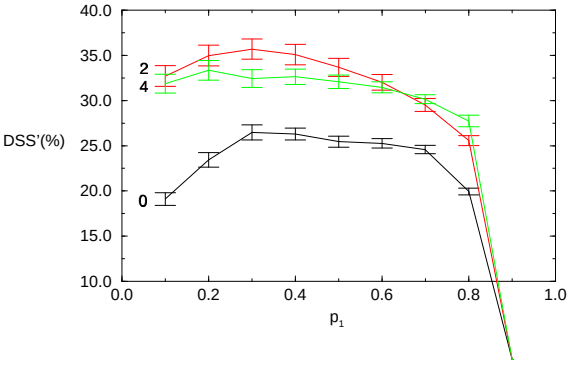

GSS(%)
Number of 0's in training set
2
4
0

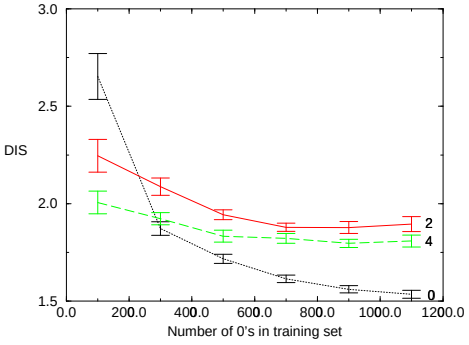

GSS'(%)
p1
4
2
0

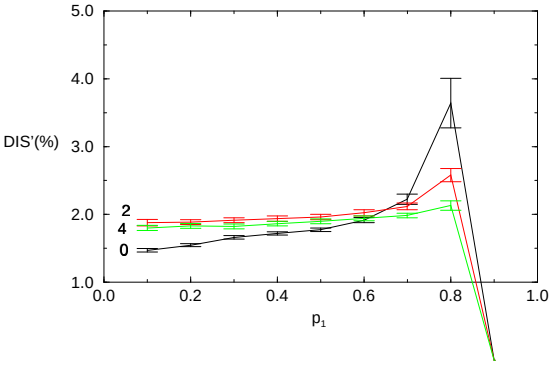

CSS(%)
Number of 0's in training set
2
4
0

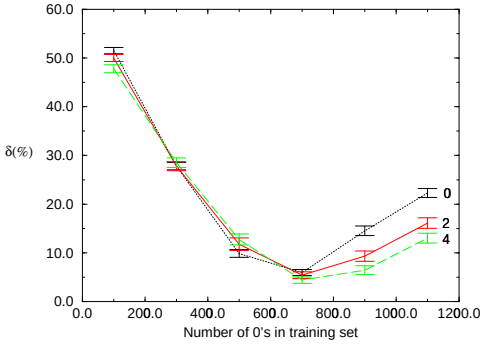

CSS'(%)
p1
2
4
0

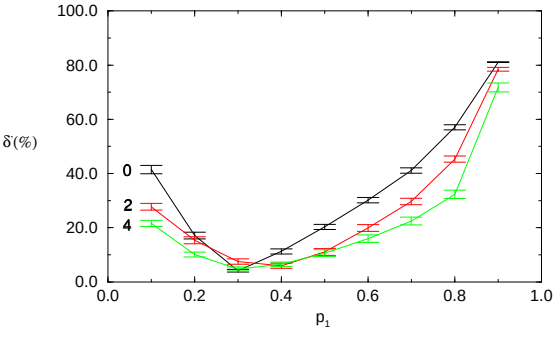

DSS(%)
Number of 0's in training set
2
4
0

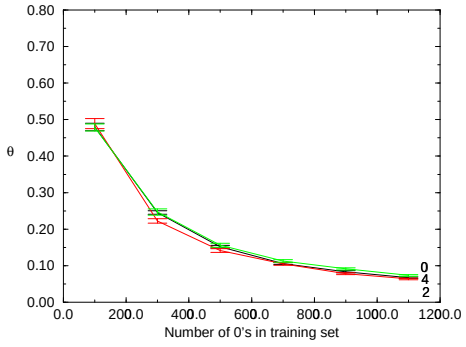

DSS'(%)
p1
2
4
0

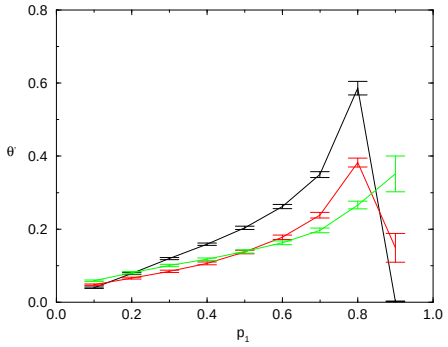

DIS
Number of 0's in training set
2
4
0

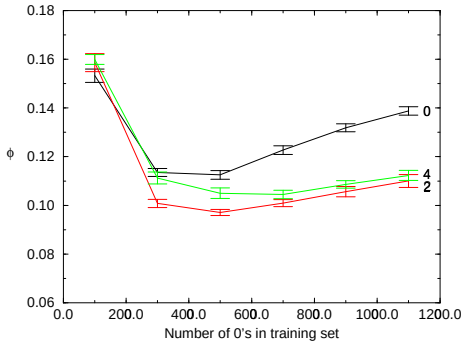

DIS'(%)
p1
2
4
0

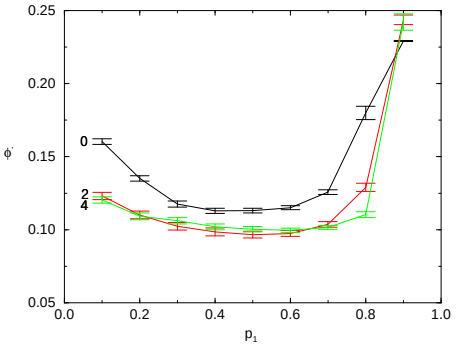

δ(%)
Number of 0's in training set
0
2
4

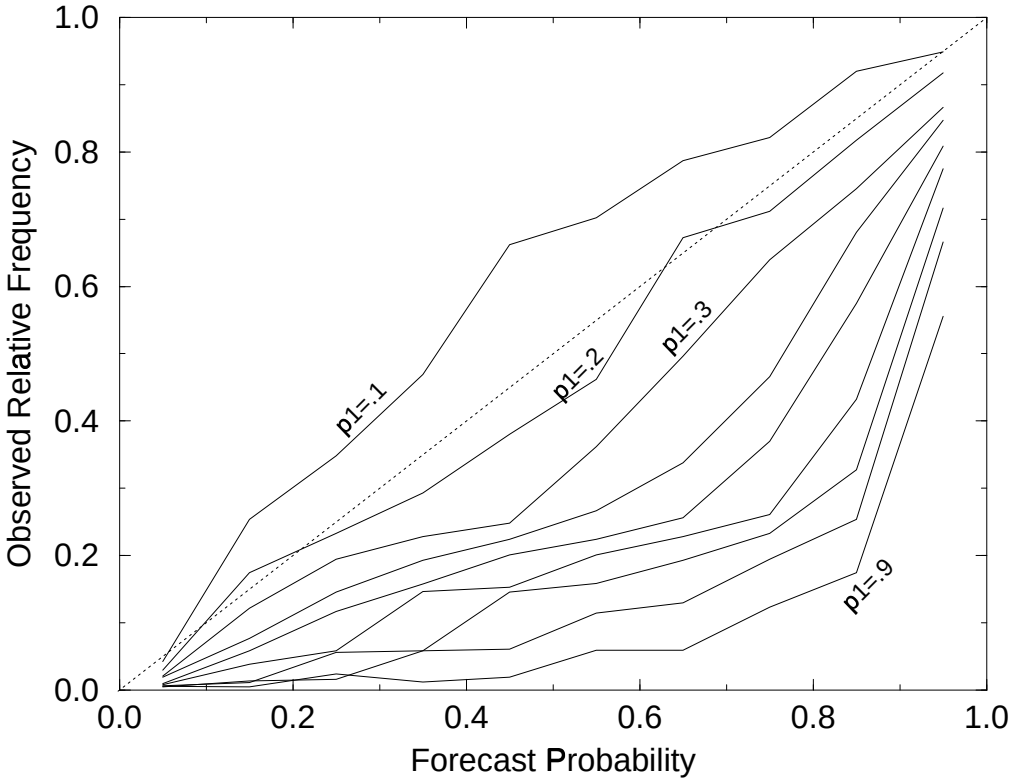

δ'(%)
p1
0
2
4

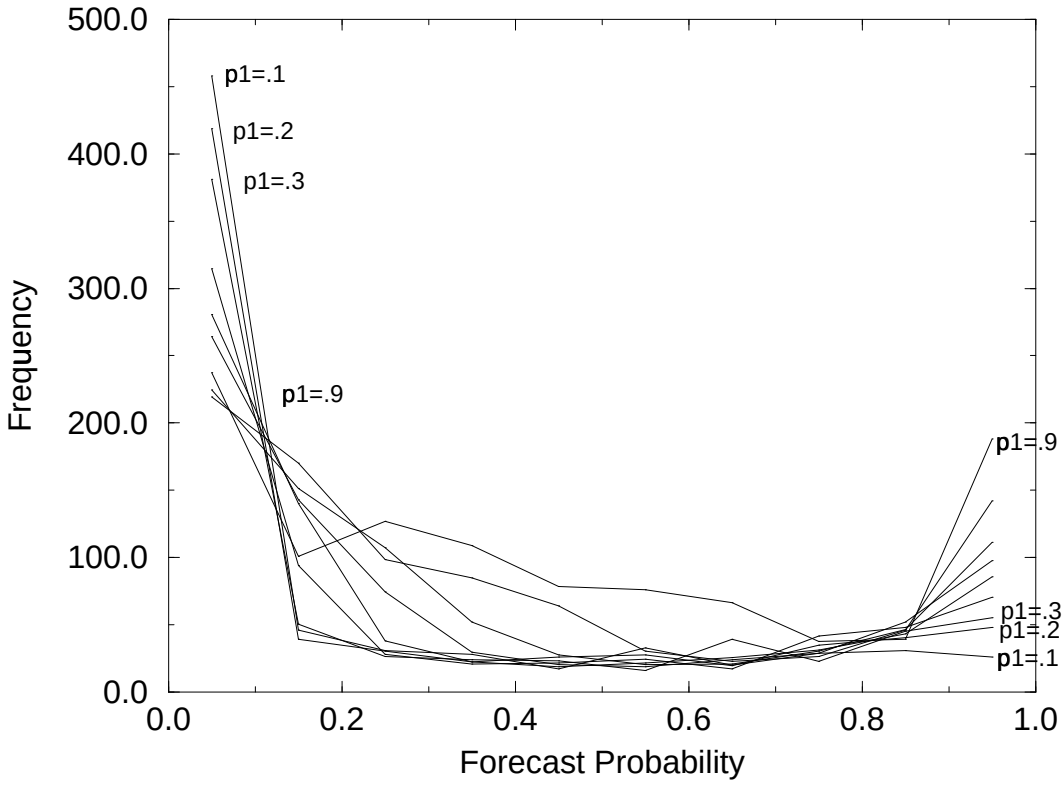

θ
Number of 0's in training set
0
4
2

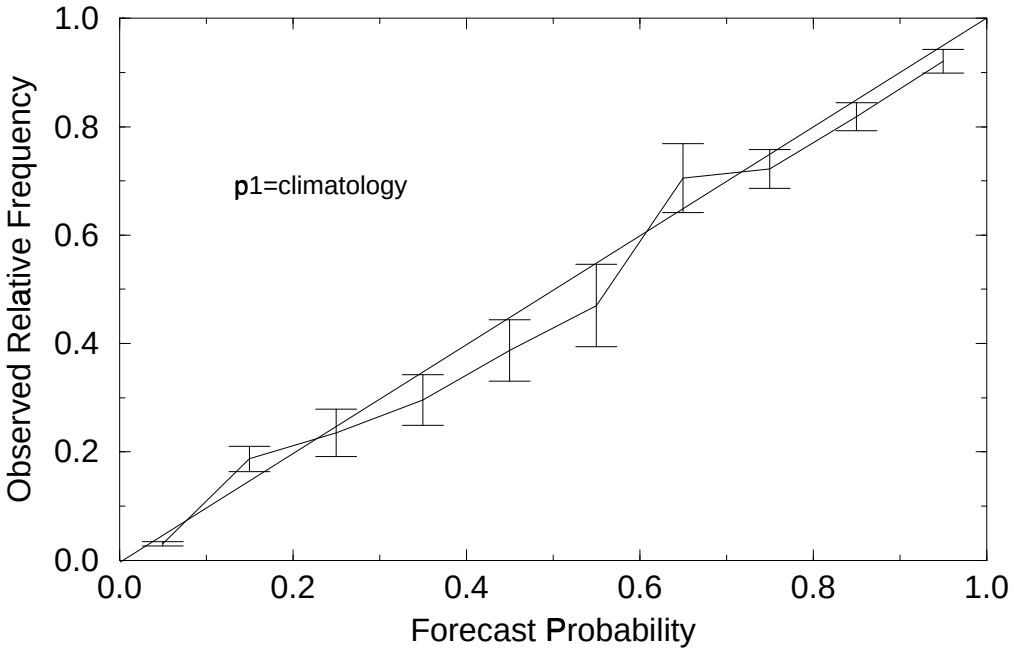

θ
p1

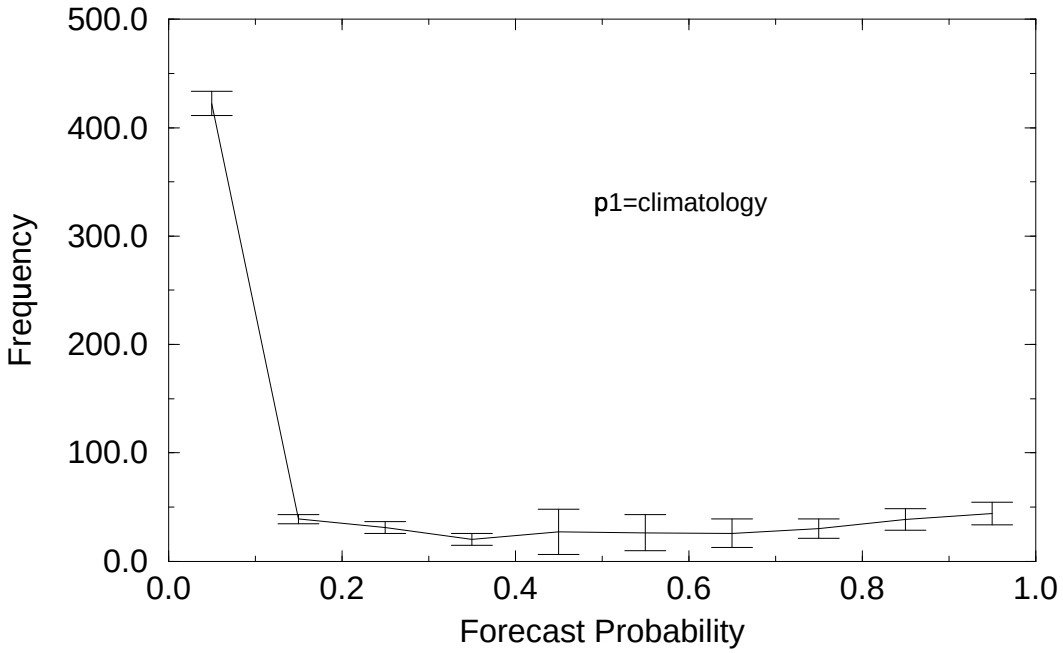

ϕ
Number of 0's in training set
0
4
2


ϕ
p1
0
2
4


Observed Relative Frequency
Forecast Probability
p1=.1
p1=.2
p1=.3
p1=.9
0.0
0.2
0.4
0.6
0.8
1.0


Frequency
Forecast Probability
p1=.1
p1=.2
p1=.3
p1=.9
p1=.9
p1=.3
p1=.2
p1=.1
0.0
100.0
200.0
300.0
400.0
500.0
0.0
0.2
0.4
0.6
0.8
1.0


1.0
0.8
0.6
0.4
0.2
0.0
Observed Relative Frequency
p1=climatology
0.0
0.2
0.4
0.6
0.8
1.0
Forecast Probability


500.0
400.0
300.0
200.0
100.0
0.0
Frequency
p1=climatology
0.0
0.2
0.4
0.6
0.8
1.0
Forecast Probability