

Calibrating a neural network-based urban change model for two metropolitan areas of the Upper Midwest of the United States

BRYAN C. PIJANOWSKI^{†*},
SNEHAL PITHADIA[‡], BRADLEY A. SHELLITO[‡] and
KONSTANTINOS ALEXANDRIDIS[†]

[†]195 Marsteller Street, Department of Forestry and Natural Resources, Purdue University, West Lafayette, IN 47907, USA

[‡]Department of Geography, 1 University Plaza, Youngstown State University, Youngstown, OH 44555, USA

We parameterized neural net-based models for the Detroit and Twin Cities metropolitan areas in the US and attempted to test whether they were transferable across both metropolitan areas. Three different types of models were developed. First, we trained and tested the neural nets within each region and compared them against observed change. Second, we used the training weights from one area and applied them to the other. Third, we selected a small subset (~1%) of the Twin Cities area where a lot of urban change occurred. Four model performance metrics are reported: (1) Kappa; (2) the scale which correct and paired omission/commission errors exceed 50%; (3) landscape pattern metrics; and (4) percentage of cells in agreement between model simulations. We found that the neural net model in most cases performed well on pattern but not location using Kappa. The model performed well only in one case where the neural net weights from one area were used to simulate the other. We suggest that landscape metrics are good to judge model performance of land use change models but that Kappa might not be reliable for situations where a small percentage of urban areas change.

Keywords: Neural nets; GIS; Kappa; Landscape pattern metrics

1. Introduction

We have developed a Geographical Information System (GIS) and artificial neural network (hereafter as neural net) based Land Transformation Model (Pijanowski *et al.* 2002a, b) that simulates land use change. Within the Land Transformation Model (LTM), a GIS is used to process spatial data that is then presented to a neural net. The neural net finds a numerical solution between the input (i.e. drivers of change) and output (e.g. locations of change occurring between two time periods) using non-linear functions and weights that are applied to units within the network. Neural nets are normally regarded as very powerful tools that can generalize well across datasets (i.e. it can perform well on data that it has not seen before); that it can process large amount of data, so large regions can be simulated at fairly high resolutions; and that they are not overly sensitive to errors in data (Skapura 1996).

*Corresponding author. Email: pijanowski@fnr.purdue.edu

Previous researchers (see Geist and Lambin 2002 for review) have found that developing land use change models that can generalize across regions is difficult. This is because the drivers of land use change are oftentimes complex interacting over different spatial and temporal scales (Lambin *et al.* 2003). Temporally, drivers of land use change (Lambin *et al.* 2003) can operate in a progressive and gradual manner (e.g. long term climate changes) or they can be episodic (e.g. floods, war). Causes can also be proximate (i.e. local), or they can originate exogenously (i.e. underlying causes). Quantifying all potential causes of land use change, be they demographic, technological, societal, economic, biophysical or policy, in a model is difficult given the lack of understanding of all of these factors, the generality of the drivers across regions and the lack of sufficient information to parameterize a model with these drivers of change (Ojima *et al.* 1994).

Neural nets have been found to be excellent tools that can generate generalizable models in other disciplines (e.g. Reed and Marks 1999, Skapura 1999). One approach to ensure that neural nets can produce a generalizable model is to employ a 'stop early' approach; in other words, allow the neural net to train for only few cycles. A recognized problem of stopping early is that the neural net could become 'trapped' in a local minima solution (Reed and Marks 1999) that will eventually produce poor results. Conversely, a neural net model that is trained for an excessive number of cycles could result in over fitting of data making it poorly transferable between applications.

The purpose of this paper is to: (1) provide a brief summary of neural net terms and concepts; (2) illustrate how we parameterize the LTM using a set of spatial interaction rules that are derived from GIS routines; (3) test the ability of neural nets to generalize across two large metropolitan areas in the Upper Midwest of the United States and (4) quantify how well neural nets performs across a test of generalization. We also present the results of four different methods to quantify model performance for our neural network models. We conclude with a discussion on how well the models perform and what the potential might be for neural nets for spatial modeling.

2. Background

2.1 Artificial neural networks

Neural nets are software tools designed to emulate the functionality of mammalian brains. Neural nets are hierarchically arranged layers of interconnected units that process information in a high parallel processing fashion. Each unit, called a *node* (analogous to a neuron), is connected to other units in network by a weighted connection so that each unit receives input from many nodes in the previous layer. The most common neural net is a multi-layer perceptron (MLP) which contains three types of layers: *input*, *hidden*, and *output*. Neural net algorithms calculate *weights* for input values, input layer nodes, hidden layer nodes and output layer nodes by introducing the input in a *feed forward* manner throughout the network layers. The output of every node is then computed as the function of its input; this function is referred to as the *activation function* and it can take on many different forms. A coefficient can be introduced to the activation function; this is called a *bias*. Weights in a neural net are determined by using a training algorithm, the most popular of which is the back propagation (BP) algorithm. The BP algorithm randomly selects the initial weights, then compares the calculated output for a given

observation with the expected output for that observation. The difference between the expected and calculated output values across all observations is summarized using the mean squared error. After all observations are presented to the network, the weights are modified according to a generalized *delta rule* (Rumelhart *et al.* 1986), so that the total error is distributed among the various nodes in the network. This process of feeding forward signals and back-propagating the errors is repeated iteratively with each iteration called a *cycle*.

Presenting data to the neural net with input and output data over many cycles is called *training*. Users can then stop the training session and have the neural net software save all of the weights and biases to a *network file*. A network file is then applied to another dataset containing only input data and no output data. This process, called *testing*, allows the neural net to estimate output values.

2.2 GIS based procedures for the land transformation model

The LTM is similar to other reduced form statistical models (e.g. Pontius 2001, Veldkamp and Lambin 2001, Walker 2003) that are grid based. Like these other models, a GIS is used calculate a variety of spatial relationships (which we call spatial transition rules) between drivers of change (e.g. road, slope, soil type) and cells that could undergo change. However, we recognize several broad classes of spatial transition rules (see Pijanowski *et al.* 2000 for details): (1) neighborhoods or densities; (2) patch size; (3) site specific characteristics; and (4) distance from the location of a driver cell. Neighborhood effects are based on the premise that the composition of surrounding cells within a certain area has an effect on the tendency of a central cell to transition to another use. Patch sizes is the size of area containing the same values (e.g. a forest patch). Site-specific characteristics are values, like slope, specific to each location. The distance spatial transition rule relates the effect of distance between each cell and the closest driver cell.

Grids that were created from the spatial transition rules are referred to as *driving variable grids*. We exclude areas that cannot undergo change; all of these cells are contained in a single GIS layer which we call the *exclusionary zone*.

3. Methods and materials

3.1 Study sites

Data for the Minneapolis-St. Paul Metropolitan area were obtained from the University of Minnesota Remote Sensing Laboratory, the Twin Cities Metropolitan Council, the Minnesota data, Vol.1 produced by the Land Management Information Center and from the Minnesota Department of Natural Resources Data Deli web site (<http://deli.dnr.state.mn.us>). Data on land use, transportation, natural features (e.g. locations of rivers, lakes, etc.), public lands, digital elevation and political boundaries were incorporated into the Arc/Info 8.X Geographic Information System (ESRI 1999). GIS data were converted from their original projection into an Albers Equal Area projection (Datum NAD 83) with units in meters.

The 1991 and 1997 Generalized Land Use data set encompass the seven county Twin Cities Metropolitan Area (TCMA) in Minnesota. The land use dataset was developed by the Twin Cities Metropolitan Council, a regional governmental organization that deals, in part, with regional issues and long range planning for the

Minneapolis-St. Paul area. The dataset (see Sawaya *et al.* 2001 for a detailed description) includes the following land use classifications: single family residential, multi-family residential, commercial, industrial, public and semi-public, airports, parks and recreation, vacant and agricultural, major four lane highways, open water bodies, farmsteads, extractive, public industrial, industrial parks not developed, and public and semi-public not developed. Land use was interpreted from 1:24000 aerial photography scanned with 0.6 m resolution pixels. Most lines were digitized at an on-screen scale of no higher than 1:3000. In highly urbanized areas, 1:1500 was more common. The Metropolitan Council attempted to meet the National Mapping Accuracy Standards at 1:24000 (within approximately 40 feet of actual location) although no testing has been conducted to verify this. For the purposes of the modeling exercises, all TCMA data were converted to an Arc/INFO GRID (ESRI 2000) format with cell sizes of 30m × 30m.

Detroit Metropolitan Area (DMA) data were obtained from several sources. Land use and transportation data from 1980 were acquired from Michigan State University's Center for Remote Sensing and Geographic Information Science. Land use interpretation was made from 1:24000 colour-infrared and black-and-white aerial photographs by the Michigan Department of Natural Resources. These data depict approximately 52 categories of urban, agricultural, forest, wetland, and other land cover types for the entire state of Michigan. Each land use/cover category is depicted by a polygon and identified with a land cover code. The minimum digitized polygon size was 5 acres. Updated land use and transportation data were obtained from the South East Michigan Council of Government's (SEMCOG) GIS facility. Land use and another GIS data used for the modeling exercise was converted to an Arc/INFO GRID (ESRI 2000) format, with 30m × 30m cell sizes.

All land use classes for both metropolitan areas were also reclassified from their original classification to Anderson Level I (Anderson *et al.* 1976) for the modeling exercises. The resulting land use/cover classes were: urban, agriculture, open grassland, forest, open water, wetlands and barren.

Figure 1 shows the areas in the Minneapolis-St. Paul (i.e. Twin Cities) Metropolitan Area and the Detroit Metropolitan Area in relationship to the Upper Midwest of the United States. Figure 2 shows land use patterns for both areas in 1978 and 1991 for DMA and TCMA respectively. Both areas are composed of 7 counties. For the TCMA study area, a total of 11.8 million cells were included in the modeling exercise. The DMA area, on the other hand, is nearly 55% larger, contained 33.3 million cells. The study interval was 7 and 15 years, for the TCMA and DMA areas, respectively.

3.2 LTM GIS parameterization

To insure the input files for the neural nets were identical for each of the two metropolitan areas, the same spatial features (e.g. highways) and spatial transition rules (e.g. distance to nearest county road) were applied to each area. The driving variable grids developed for our models summarized below:

3.2.1 Transportation. The absolute distance each cell in the entire location from the nearest a) highway and b) county road or residential streets, was stored in

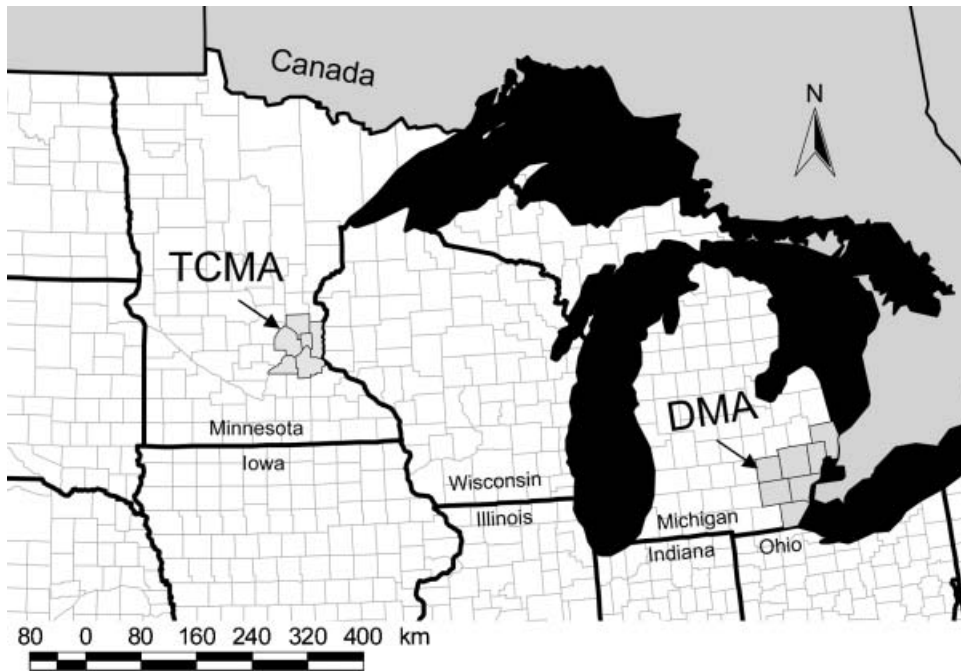


Figure 1. Location of the two metropolitan areas, the Twin Cities Metropolitan Area (TCMA) and the Detroit Metropolitan Area (DMA). State (in black) and county (grey) boundaries are shown.

separate Arc/INFO GRID files. These two driving variable grids represented the potential accessibility of a location for new development. A third transportation grid, distance from downtown, was used to reflect the distance between housing and the greatest concentration of employment opportunities for the metropolitan area.

3.2.2 Landscape features. The distance from lakes and rivers was calculated and also stored as separate driving variable grids in the GIS. Pijanowski *et al.* (2002a) has found that landscape topography is an influential factor contributing toward residential use. Thus, a 'rolling hills' driving variable grid was created from a 30m Digital Elevation Model (DEM). The amount of topographic variation surrounding each cell was estimated by calculating the standard deviation of all cell elevations within a 5km square area. Cells containing larger values reflect landscapes that contain a greater amount of topographic relief around them.

3.2.3 Urban services. The distance each cell was from the nearest urban cell during the start the model (TCMA=1991, DMA=1980) was calculated and stored as separate driving variable grids. It is assumed that the costs of connecting to current urban services (e.g. sewers) decrease with distance from urban.

3.2.4 Exclusionary zones. In the TCMA, the following areas were held back as being areas of non-development: local parks, existing urban areas, water, and public lands (including national wildlife refuges, national forests, state forests, and state parks). In the DMA, the following areas were considered areas of non-development: existing urban areas, water, and designated public lands.

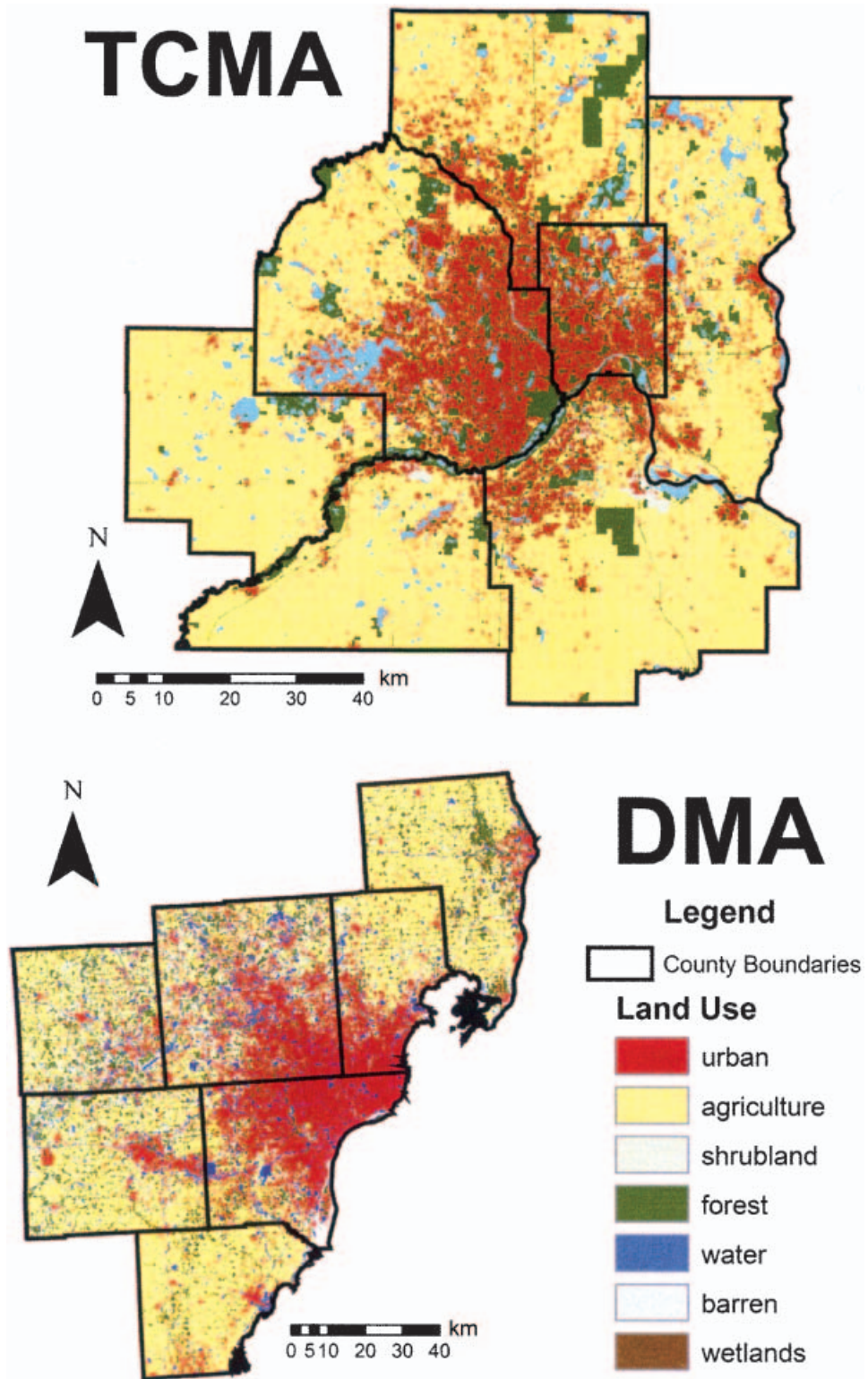


Figure 2. Land use maps of the two study areas.

3.3 Neural network parameterization

Driving variable grids stored in the GIS were written to ASCII grid files and then converted to a tabular format such that each location contained its spatial configuration value (i.e. each location was an input vector into the neural net) from each driving variable grid. This reformatting to a tabular arrangement was necessary for input to the neural net software (see below for more details). The neural network model was based on Pijanowski *et al.* (2002a). Briefly, each value in an entire driving variable grid was normalized from 0.0 to 1.0 by dividing each value by the maximum value contained in driving variable grid. Cells located within the exclusionary zone were removed.

The Stuttgart Neural Network Simulator (SNNS) was utilized for training and testing. To reduce the possibility that the neural net would overtrain, every other cell was presented to the neural net. The SNNS *batchman* utility was used to create, train and test the neural net. A back propagation, feedforward neural net, with one input layer, one hidden layer and one output layer was utilized. The neural network input layer contained seven nodes (one node for each driving variable) and seven nodes in the hidden layer. The output layer contained binary data that represented whether a cell location changed to urban (1=change; 0=no change) during the study period (1980 to 1995 for DMA and 1991 to 1997 for TCMA).

We allowed the neural net to train on the input and output data for 1000 cycles and saved the network file at 100 cycle intervals. To reduce the possibility that the network would bias its learning based on the order of data presented to it, the *shuffle* option in SNNS was used to randomly present data during each training cycle.

The testing exercise that followed used driving variable grids from all cells (except those located in the exclusionary zone) in the study locations but with the output values removed. The network file generated from the training exercise was used to estimate output values for each location. The output was estimated as values from 0.0 (not likely to change) to 1.0 (likely to change); the output file created from this testing exercise is called a *result file*. Values from the result file were multiplied by 100 to produce values ranging from 0 to 100; we call these *suitability* values. We then used a maximum likelihood rule (see Pijanowski *et al.* 2002a for details) such that cells with the greatest values were assumed to transition first. The number of cells selected to transition equaled the number of cells transitioning between the two years being modeled. A routine was written into the C programs that performed this calculation such that if cells possessed the same value but only a subset of them needed to be selected to transition, then the necessary number of cells of equal value was randomly selected from the pool of available cells. It should be noted that in logistic models it is common to threshold those cells whose value is 0.5 or greater (see Pontius 2002). However, output values from a neural net can not be directly interpreted as probabilities (Reed and Marks 1999), especially when output data are highly unbalanced as is the case with both of our study areas; in many instances the greatest value from a neural net simulation was less than 0.5.

To test how well the neural net model could be applied to other areas, we performed a *swap* of network files between the two metropolitan areas. For example, one type of swap involved training the neural net on DMA data and applying (i.e. testing) the network file to TCMA data to derive output. We ran the models for 1000 cycles saving model simulation results for every 100 cycles.

In order to determine how well neural nets perform for an extremely large number of cycles, we selected a subset of the TCMA that was 1% of the metropolitan area (containing 10810 cells). This area also had 50% of the area transition to urban. We used the identical inputs that were subset using the GIS but ran the simulation out for 1 000 000 cycles. The network files were saved every 1000 cycles (although we report results on a 100 cycle run as well). We repeated the accuracy assessment analysis for this subset and report the results of selected simulations for comparison purposes.

3.4 LTM accuracy assessment

We used a variety of methods to assess model performance. First, cells that were predicted to transition to urban (according to the model output) were compared with the cells that actually did transition during the time period of study. The percentage of cells falling into this category was then divided by the actual number of cells transitioning to obtain a percent correct match (PCM) metric, calculated as follows:

$$\frac{\# \text{ cells correctly predicted to change} \times 100.0}{\# \text{ cells actually transitioning}} \quad (1)$$

Following our earlier approach (Pijanowski *et al.* 2002a), we passed a scaleable window through what we call the error grid containing locations of correctly predicted cells and omission and commission errors. We increased the numerator in equation (1) by one for each omission and commission error pairs falling in this window (we did not count any error cell more than once). We first passed a 3×3 window through the entire error grid starting with the upper left corner. The final adjusted PCM was then saved to a file. We next increased the window by two and repeated the process. Windows sizes between 3×3 and 99×99 were used to calculate adjusted PCMs for swap and non-swap simulations. This approach allows us to estimate the accuracy across different scales (i.e. different window sizes). We report here the window size where the PCM exceeds 0.5 (i.e. the window size where the number of correctly predicted cells exceeds the number not correctly predicted after pairs of omission and commission errors are counted in the window). An example of output from our scaleable window with the PCM threshold value of 50% is illustrated in figure 3.

Our second approach to assess model performance was to determine how well the model output matched the real change maps. We calculated a Kappa statistic (Sousa *et al.* 2002) for each model run. Kappa calculates the percentage success of a model relative to chance (table 1).

We used the following form of the Kappa equation:

$$K = \frac{P(A) - P(E)}{1 - P(E)} = \frac{\sum_{i=1}^c p_{ii} - \sum_{i=1}^c p_{iT} \cdot p_{Ti}}{1 - \sum_{i=1}^c p_{iT} \cdot p_{Ti}} \quad (2)$$

where,

$i=1, \dots, c$: the common categories of change in both observed change and simulated model run results. There are two categories (i.e. $c=2$)

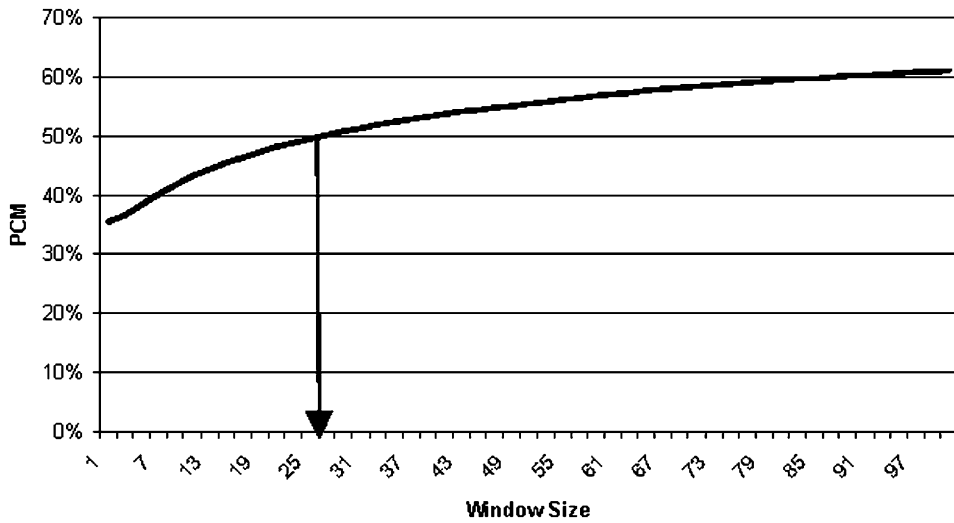


Figure 3. The scaleable window analysis plot illustrating the window size that produces a PCM of 50%.

categories whose values are predefined ‘0’ (no urban change) and ‘1’ (urban change).

p_{ij} : the proportion of cells in category i of observed change in category j of simulated model run, from the contingency or confusion matrix.

p_{iT} : the proportion of cells in category i of observed change, taken from the marginal totals of the last column of the contingency matrix.

p_{Ti} : the proportion of cells in category i of simulated model run, taken from the marginal totals of the last row of the contingency matrix.

p_{ii} : the proportion of cells in the same category, i , on both observed changes and simulated model runs, taken from the diagonal elements of the contingency matrix.

$P(A) = \sum_{i=1}^c p_{ii}$: the fraction of agreement, or sensitivity coefficient.

$P(E) = \sum_{i=1}^c p_{iT} \cdot p_{Ti}$: the expected fraction of agreement subject to the observed distribution.

It is generally considered (Sousa *et al.* 2002) that Kappa values for map agreement are: >0.8 is excellent; 0.6–0.8 is very good; 0.4–0.6 is good; 0.2–0.4 is poor; <0.2 very poor.

We were also interested in how well the models predicted patterns of urban change compared to patterns of real urban change (hereafter referred to as *real change*) that occurred in the two metropolitan areas. We used the landscape metric program FRAGSTATS 3.3 (McGarigal *et al.* 2002) to calculate eight different landscape pattern metrics (see table 2) on grids with cell values of 1 for model predicted change and real change. We chose three patch metrics (number of patches, mean patch area and standard deviation of patch areas), one shape metric (mean patch fractal dimension), one aggregation metric (landscape shape index) and one connectivity metric (patch cohesion index). All landscape

Table 1. Summary of the landscape pattern metrics from FRAGSTATS used for model performance assessment.

Name	Equation	Description & Arguments
Number of Patches (NP)	$NP = n_i$	n_i = number of patches in urban change class. <i>Range</i> : $NP \geq 1$. When $NP = 1$, the landscape contains only one patch of the urban change. Measures the extent of subdivision or fragmentation of the urban change patch. Since across the model runs the total landscape area and the number of cells that are transitioned in each run are held constant, the number of patches conveys information about the patch density and mean size. The metric produces the number of patches of the corresponding patch type (urban change).
Mean Patch Area (AREA_MN)	$A_{mean} = \frac{\sum_{j=1}^n a_{ij} \left(\frac{1}{10,000}\right)}{n_i}$	n_i = number of patches in urban change class, a_{ij} = area of patch ij in square meters, divided by 10,000 to be converted in hectares. The final metric display is expressed as a percentage change in mean patch area from the real urban change mean patch area observed. Measures the mean area of each urban patch comprising the model landscape mosaic for each run.
Patch Area Standard Deviation (AREA_SD)	$A_{SD} = \sqrt{\frac{\sum_{j=1}^n \left(a_{ij} \left(\frac{1}{10,000}\right) - \left(\frac{\sum_{j=1}^n a_{ij} \left(\frac{1}{10,000}\right)}{n_i} \right) \right)^2}{n_i}}$	Similar to the Mean Patch Area metric, this quantifies the root square error, that is the deviation from the mean, of patch area in the corresponding urban change grid
Landscape Shape Index (LSI)	$LSI = \frac{e_i}{\min e_i}$	e_i = total length of edge of class i (urban change) in terms of number of cell surfaces; $\min e_i$ = minimum total length of edge of class i (urban change) for a maximally aggregated urban achieved if all urban was aggregate into one square patch. <i>Range</i> : $LSI \geq 1$. $LSI = 1$ when the landscape consists of a single square.
Mean Fractal Dimension Index (FRAC_MN)	$FRAC_{mean} = \frac{\sum_{j=1}^n \frac{2 \ln(0.25 p_{ij})}{\ln a_{ij}}}{n_i}$	p_{ij} = the perimeter (in m), and a_{ij} = the area (in m^2) of urban patch ij . <i>Range</i> : $1 \leq FRAC_{mean} \leq 2$, indicating the extend of departure from Euclidian geometry or increase in shape complexity. This metric measures the complexity of shapes across a range of spatial scales defined by the urban patch sizes. For very simple urban patch types (e.g., squares), the metric approaches 1, while for more complex type of urban patches, the metric approaches 2.
Patch Cohesion Index (COHESION)	$COHESION = \left(1 - \frac{\sum_{j=1}^n p_{ij}}{\sum_{j=1}^n p_{ij} \sqrt{a_{ij}}} \right) \left(1 - \frac{1}{\sqrt{A}} \right)^{-1} \quad (100)$	p_{ij} = perimeter of patch ij in terms of number of cell surfaces. a_{ij} = area of patch ij in terms of number of cells. A = total number of cells in the landscape. <i>Range</i> : $0 \leq COHESION \leq 1$. $COHESION = 0$ when the proportion of landscape comprised of the urban class decreases and becomes increasingly subdivided and less physically connected. Measures the physical connectedness of the corresponding patch type.

Table 2. Confusion matrix for model performance showing row and column units important for the Kappa coefficient calculation.

Observed Change	Simulated Model Runs		
	0	1	Total
0	p_{11}	p_{12}	p_{1T}
1	p_{21}	p_{22}	p_{2T}
Total	p_{T1}	p_{T2}	1

metrics were compared to the real change metrics as a percent difference from real change:

$$\frac{LM_{rc} - LM_m}{LM_{rc}} * 100 \tag{3}$$

Where LM is one of the above six landscape metrics, m represents the LM for a model simulation and rc is the LM for real change grid.

4. Results

4.1 Summary of land use changes

Within the TCMA, urban use was 21% of the entire study area in 1991 and 23% in 1997, which represents a total of 249 071 cells transitioning to an urban land use in the TCMA during the 6-year period. On the other hand, urban use in 1980 comprised 26% of the entire area in the DMA. In 1995, urban was 33% of the DMA study area with 744 078 cells transitioning to urban in the DMA during the 15 year time period. Thus, for TCMA, about 4.87% of the area converted to urban between 1991 and 1997 and 6.88% converted to urban in DMA from 1980 to 1995.

4.2 Scaleable window

Figure 4 shows the results of the scaleable window analysis of the TCMA and DMA simulations including the swaps for simulation cycles 100 to 1000 with 100 cycle

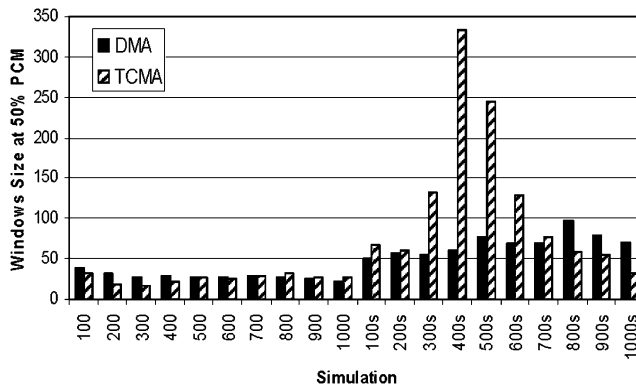


Figure 4. Window sizes for scaleable window analysis that gives a PCM of 50%. Simulations for TCMA and DMA non-swap are shown for 100-1000 cycles at 100 intervals. The swaps (denoted with an s) are shown for the same intervals as the non-swap simulations.

increments. Note that for the non-swap simulations, both TCMA and DMA produce a PCM greater than 50% at window sizes between 17 and 39. This represents windows of that are 0.510 km to 1.17 km across. The swaps, on the other hand, have window sizes as large as 333 cells wide, representing a window size of 9.99 km. Only the TCMA swap of 1000 cycles performed as well as the training and testing on itself. In other words, the best swap was when the network file from the DMA training of 1000 cycles was applied to the TCMA input data. It is also interesting to note that the window size of 50% PCM for TCMA swaps start small, around 67 for 100 cycles, steadily rises peaking at 400 cycles and then falls again to its lowest level at 1000 cycles which had a window size of 31 (0.930 km) where PCM reached 50%. The DMA swap simulations produced more even results for window sizes for 50% PCM fluctuating between 50 (1.5 km) and 97 (2.9 km).

4.3 Percentage map agreement

We took the output of predicted change and calculated the percentage of cells that were similar between model simulations. Figure 5 shows the results of pairwise comparisons of model output and the percentage of cells that were similar for three different cycle ($A=100$, $B=500$ and $C=1000$) simulations. For TCMA, nearly all of the pairwise comparisons showed that there was at least an 80% agreement in the locations of cells that were predicted to change. Nearly 78% of the cells were the same in all three runs (i.e. $A+B+C$). There was much less consistency between model results for the DMA area with pairwise runs producing 55 to 73% percent agreement. Only 51% of cells were the same for all three model runs for DMA.

4.4 Kappa statistic

Kappa coefficients for all simulations are shown in figure 6. The range for Kappa is 0.12 to 0.30. Nearly all of the non-swap TCMA runs produced the same Kappa (~ 0.29); the DMA simulations produced more variable Kappas across these simulations. Kappas for TCMA exceed those for DMA. The Kappas for all swap

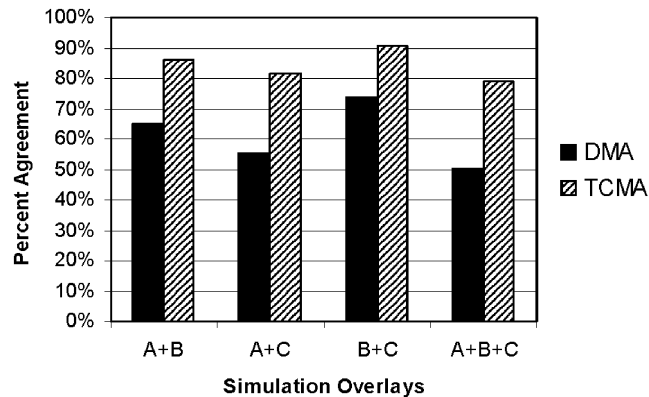


Figure 5. The percent agreement between pairs of model simulations and all three simulations for 100 (A), 500 (B) and 1000 (C) cycle runs.

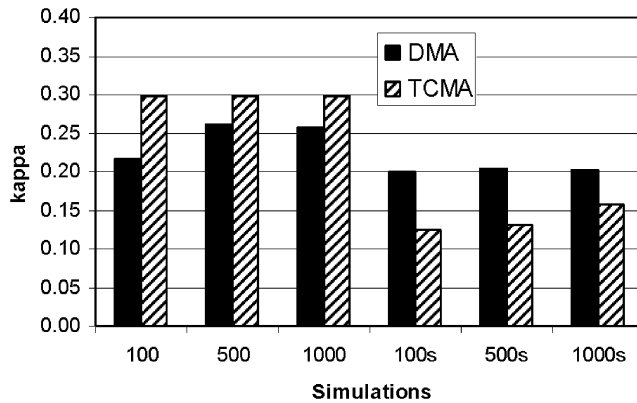


Figure 6. Kappas for DMA and TCMA non-swap and swap (denoted with an *s*) simulations for 100, 500 and 1000 cycle runs.

simulations were lower than non-swap with the TCMA swap (i.e. DMA trained network file applied to TCMA input) yielding the smallest Kappas.

4.5 Landscape metrics

Figure 6 summaries the landscape metric analysis performed on 100, 500 and 1000 cycle simulations for non-swap and swap simulations. The number of patches (NP) did not differ greatly from the DMA real change grid for all non-swap simulations. The non-swap TCMA simulations did produce more patches for 100 and 1000 cycle simulations; as many as 47% more than the number of patches in the real change grid for TCMA. In addition, the swap simulations produced more patches in both areas compared to the number of patches in the real change.

The mean area (AREA_MN) of patches in non-swap DMA simulations did not differ greatly (<10%) from the mean patch area in the real change. However, the mean patch area for the TCMA non-swap simulations was greatly smaller with patch areas of predicted urban being less than half of the mean patch area for the real change grid in TCMA. For the swap simulations, both areas produced mean patch areas that were much smaller than the real change for the simulated area.

The standard deviation of the patches (AREA_SD) of predicted urban for DMA and TCMA were less than the metric for real change in all swap and non-swap simulations. This suggests that the model is producing less variable patches of urban than what occurs in both study area's real change grid.

The landscape shape index (LSI) measures the total length (in cell units) of edge of urban. Thus, LSI measures the amount of urban aggregation or clumpedness. The LSIs for the swap and non-swap simulations varied greatly. The DMA non-swap simulations produced LSI differences that were smaller than the real change. The largest difference was for the 500 cycle TCMA non-swap simulation.

The mean fractal dimensions of urban patches were smaller for all simulations, swap and non-swap. FRAGSTATS calculates a fractal dimension for each patch varying between 1 (a patch with a very simple perimeter) and 2 (a patch with shapes containing highly convoluted perimeters). These values did not differ greatly though from the real change grids; the largest percent difference from the real change slightly exceeded 5%.

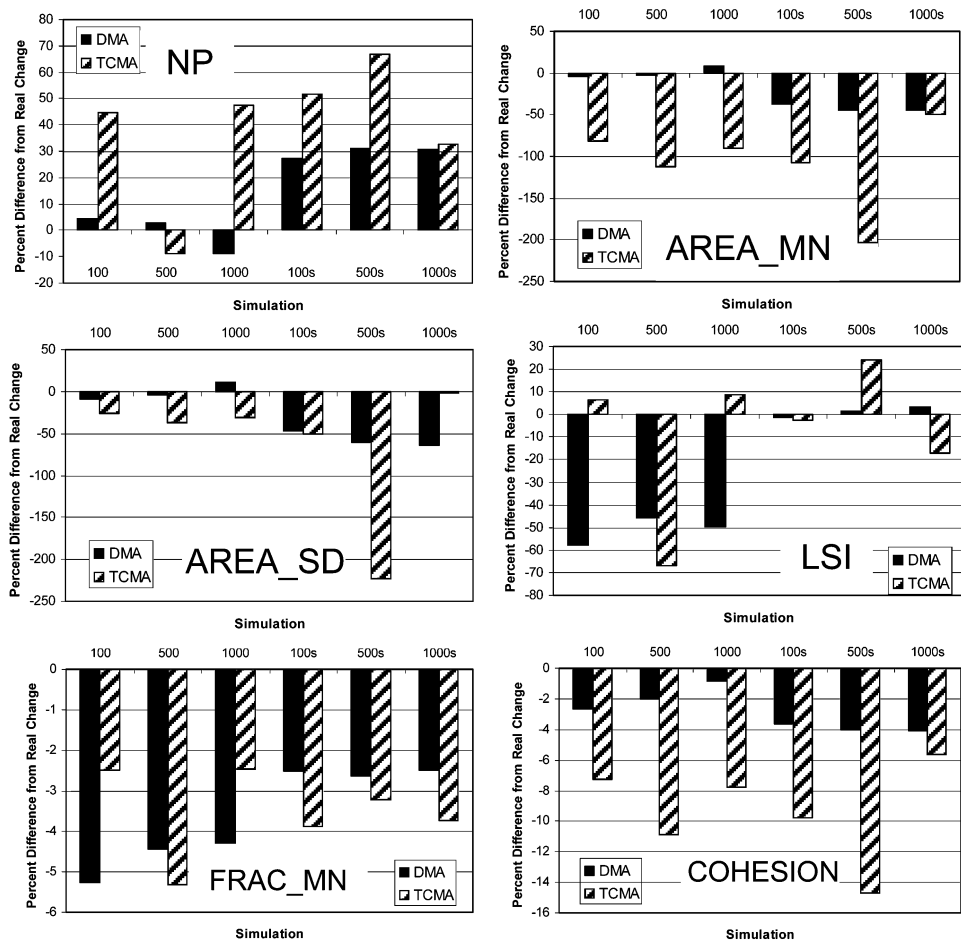


Figure 7. Landscape pattern metrics for TCMA and DMA non-swap and swap runs expressed as a percentage difference from real urban change map. See table 1 for details of landscape equations and descriptions.

4.6 Subset simulations

We saved the result file containing the suitability values for all simulations between 1000 and 1 000 000 cycles at 1000 intervals (in addition to the 100 cycle run). The suitability values were then multiplied by 100 and their integer values saved to a file. We then plotted (figure 8) the number of cells in of the 100 suitability values for several of the simulations across this range of 1000 different simulations. It is interesting to note how the neural nets first produce suitability values that are aggregated toward 0; at 5000 cycles the mean value in suitability values approach 50 but at larger cycle simulations, greater than 100 000 cycles, the neural net begins to produce a large distribution of suitability value with the most frequent values being at the extremes. This suggests that the neural net at 1 000 000 cycles has more confidence in many cells which are either 0 (no change) or 1 (change). In addition, the middle portion of this distribution, which likely represents ambiguity in change and no change, becomes smaller.

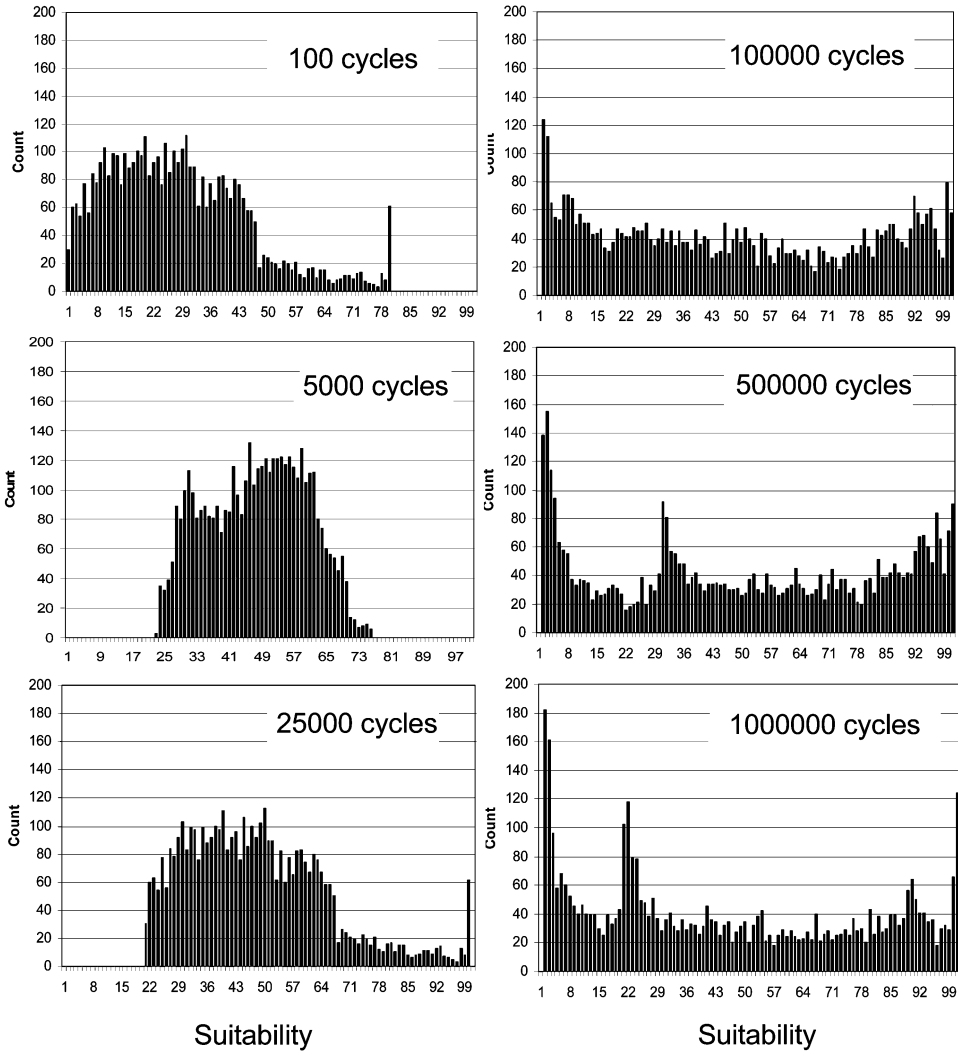


Figure 8. Suitability distributions for selected runs of cycles between 100 and 1000 000. Suitability values are neural net output values multiplied by 100 and truncated to integers.

We calculated Kappas for a wide range of simulations for this subset as well (figure 9). Kappas start at 0.3 for the 100 cycle simulation, dip to 0.23 for the 5000 cycle simulation, and then increase with increasing number of cycles until it passes 0.5 around the 60 000 cycle simulation. Kappa plateaus around 0.6 for 100 000 cycle runs up to the 5 000 000 cycle run.

The percent difference of each landscape metric with that of the subset real change is shown in figure 10 for the same cycle simulations as we reported for Kappas. The percent difference from real change for NP, AREA_MEAN, AREA_SD, and COHENSION metrics are all smallest for model simulations with 20 000 cycles or more. The LSI showed a different trend with this metric being larger than real change, then smaller than real change at 20 000 cycles but not differing much with real change LSI for simulations with greater than 60 000 cycles. The percent

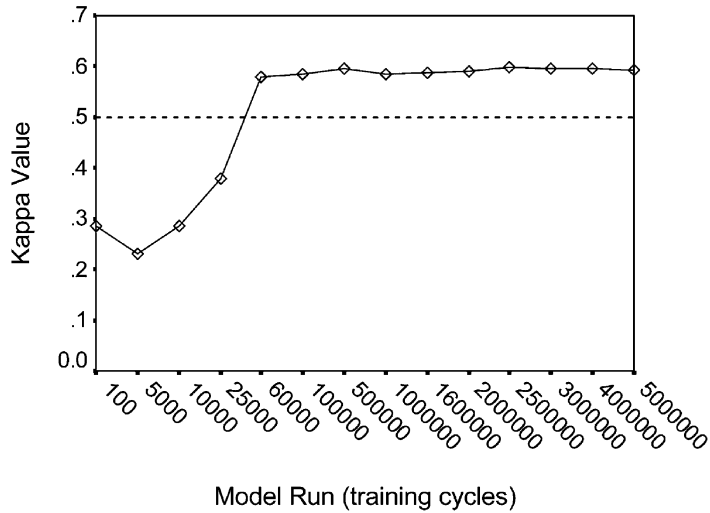


Figure 9. Kappa coefficients for TCMA subset simulations for selected runs of cycles between 100 and 1000 000.

difference from real change mean fractal dimension did not differ greatly across all model simulations.

5. Discussion

Neural nets are powerful tools that allow users to build complex models that have nonlinear relationships (Bishop 1999). It is oftentimes stated (e.g. Reed and Marks 1999, Skapura 1999) that neural nets can generalize well across data sets. One technique to ensure that neural nets can produce a generalizable model is to employ a ‘stop early’ approach; in other words, allow the neural net to train for only few a cycles. A recognized problem of stopping early, especially with back-propagation networks, is that the neural net could become ‘trapped’ in a local minima solution (Reed and Marks 1999) that will eventually produce poor results.

We parameterized the GIS and neural net-based LTM for the Detroit and Twin Cities Metropolitan Areas. We built several neural net models and attempted to test whether these models were transferable across the two metropolitan regions for neural nets that are stopped early. In past model simulations (e.g. Pijanowski *et al.* 2002a), we ran simulations for several thousand cycles and found models performing well around 4000 cycles.

We built two different types of models. The first, were models where we trained and tested using data from the same area. The second, which we refer to as swap simulations, were designed to create network files developed from a training exercise in one area and then testing (e.g. network file from the other area) on the other input dataset.

We used a variety of methods to assess model performance. This included: (1) determining how well each model predicted location across spatial scales, as reflected in our scaleable window analysis, (2) what percentage of cells in the different model simulations were the same; (3) how well model predictions of urban change differed from real change using the Kappa statistics, and (4) how well the models predicted spatial patterns compared to the spatial pattern of urban change.

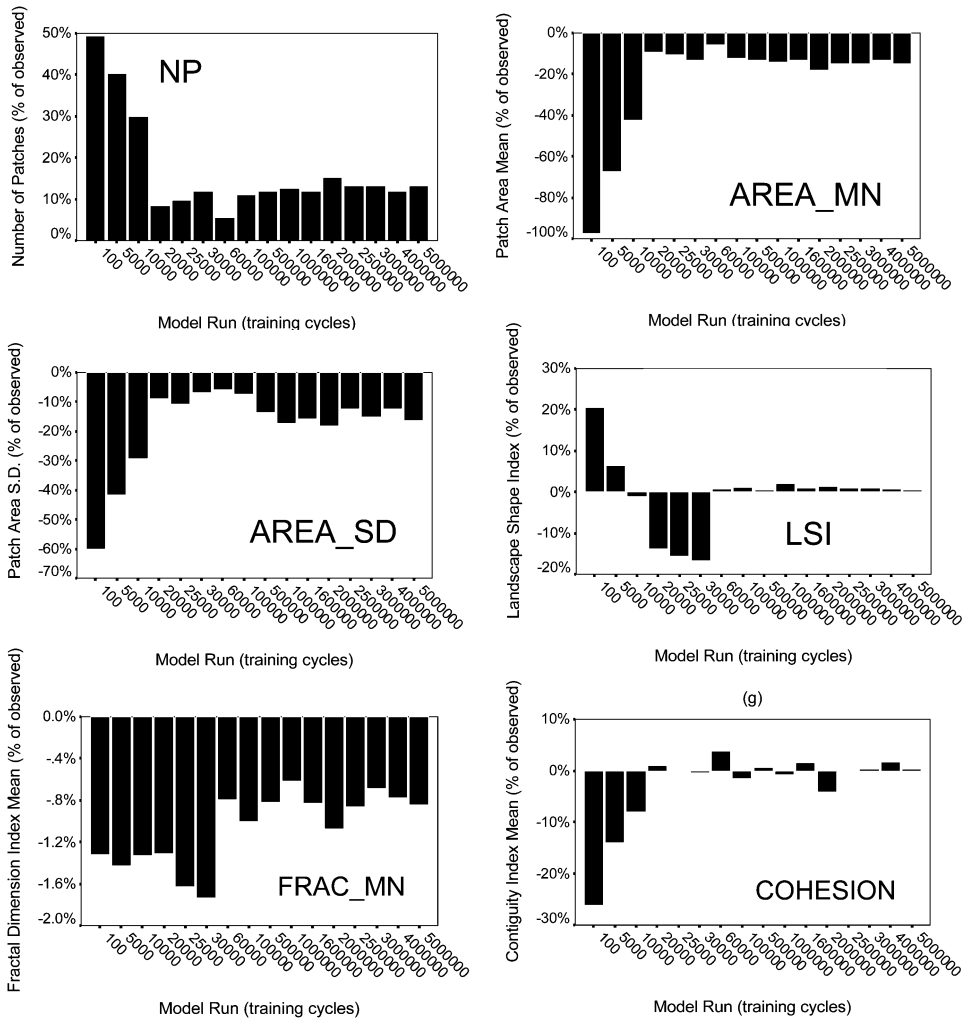


Figure 10. Landscape pattern metrics for the TCMA subset simulation for selected runs of cycles between 100 and 5 000 000. The values are expressed as percentage different from real urban change that occurred in the subset region.

Because we were also interested in how well the neural nets performed for extended training cycles, we created a subset of a model from the TCMA area that represented 1% of the study area. We trained this area for 5 000 000 cycles saving network files for every 1000 cycle iterations. This area also included a higher number of urban change percentages (around 50%) so we could evaluate the model when the neural net is presented with an equal number of change and no change cells.

For the non-swap simulations, both TCMA and DMA performed well (i.e. had PCMs greater than 50%) at a 1 km scale. The swap simulations, however, produced highly variable results; only one swap simulation (TCMA 1000 cycle) performed as well as the non-swap.

The Kappas for the TCMA and DMA non-swap and swap simulations were all very low, suggesting poorly performing models. However, according to Fielding (2002), the Kappa coefficient is sensitive to sample size and is unreliable if one class

of the data dominates in the dataset under examination (Fielding 2002: 277). For each of our metropolitan areas, urban change occurred in only 4.87% of the area in TCMA and 6.88% in DMA. Indeed, for a more balanced model simulation where the number of change and no change cells were nearly equal (i.e. our subset simulation), the Kappa exceeded the often perceived threshold for model validation of $K=0.4$ (Boone and Krohn 2002, Fielding 2002, Sousa *et al.* 2002), and $K=0.5$ (Pontius 2000, Pontius 2002).

In conclusion, we believe that neural nets show promise in producing good models of urban change. We show that the number of training cycles produces greatly differing results so care must be used in assuring that model performance be assessed across a range of training cycles. Our model performance methods suggest that a variety of techniques are useful in judging model performance. These include accuracy across scales and metrics of spatial pattern. We believe that Kappa is a reliable test for areas undergoing a lot of land use change. It may not, however, be robust for models that need to predict fairly rare occurrences of change.

Acknowledgements

This work was supported in part from NASA grant NAG13-99002 to the University of Minnesota, University of Wisconsin and Michigan State University entitled 'An Upper Great Lakes Regional Environmental Science Applications Center (RESAC)'. Data for the Twin Cities Metropolitan area were provided by Marvin Bauer and Kali Sawaya at the University of Minnesota, St. Paul. We also acknowledge the cooperation of the Southeast Michigan Council of Governments (SEMCOG) and the Twin Cities Metropolitan Council (TCMC) for providing land use and land features data used for the modeling exercises presented here. We also would like to thank: Gaurav Manik, Aaron Keller, and Fei Yuan who also worked on the project developing databases, conducting analyses and writing customized programs used in the modeling work. We gratefully acknowledge useful comments from three anonymous reviewers and Peter Verburg.

References

- ANDERSON, J.R., HARDY, E.E., ROACH, J.T. and WITMER, R.E., 1976, *A land use and land cover classification system for use with remote sensor data*, US Geological Survey, Professional Paper 964, p28, Reston, VA.
- BISHOP, C., 1999, *Neural networks for pattern recognition* (Oxford University Press: Oxford, England).
- BOONE, R.B. and KROHN, W.B., 2002, Modeling tools and accuracy assessment. In *Predicting species occurrences: Issues of accuracy and scale*, edited by J.M. Scott, P.J. Heglund, M.L. Morrison, J.B. Haufler, M.G. Raphael, W.A. Wall and F.B. Samson (Washington DC: Island Press), pp. 265–270.
- ENVIRONMENTAL SYSTEMS RESEARCH INSTITUTE, 2000, Arc info online documentation ArcDOC version 8.3. *Cell-based modeling with GRID*, Environmental System Research Institute technical documentation, Redlands, California.
- FIELDING, A.H., 2002, What are the appropriate characteristics of an accuracy measure? In *Predicting species occurrences: Issues of accuracy and scale*, edited by J.M. Scott, P.J. Heglund, M.L. Morrison, J.B. Haufler, M.G. Raphael, W.A. Wall and F.B. Samson (Washington DC: Island Press), pp. 271–280.
- GESIT, H. and LAMBIN, E., 2002, Proximate causes and underlying driving forces of tropical deforestation. *BioScience*, **52**, pp. 143–150.
- MCGARIGAL, K., CUSHMAN, S.A., NEEL, M.C. and ENE, E., 2002, FRAGSTATS: Spatial pattern analysis program for categorical maps. Computer software program produced

- by the authors at the University of Massachusetts, Amherst. Available at the following web site: www.umass.edu/landeco/research/fragstats/fragstats.html
- LAMBIN, E., GEIST, H. and LEPELERS, E., 2003, Dynamics of land-use and land-cover change in the tropical regions. *Annual Review of Environmental Resources*, **28**, pp. 205–241.
- OJIMA, D., GALVIN, K. and TURNER, B. II, 1994, The global impact of land-use change. *BioScience*, **44**, pp. 300–304.
- PIJANOWSKI, B.C., GAGE, S.H., LONG, D.T. and COOPER, W., 2000, A land transformation model: Integrating policy, socioeconomics and environmental drivers using a Geographic Information System. In *Landscape ecology: A top down approach*, edited by L. Harris and J. Sanderson, pp. 184–198.
- PIJANOWSKI, B.C., BROWN, D., SHELLITO, B. and MANIK, G., 2002a, Using neural networks and GIS to forecast land use changes: A land transformation model. *Computers, Environment and Urban Systems*, **26**, pp. 553–575.
- PIJANOWSKI, B.C., SHELLITO, B., PITHADIA, S. and ALEXANDRIDIS, K., 2002b, Using artificial neural networks, geographic information systems and remote sensing to model urban sprawl in coastal watersheds along eastern Lake Michigan. *Lakes and Reservoirs*, **7**, pp. 271–285.
- PONTIUS, R.G., JR, 2000, Quantification error versus location error in comparison of categorical maps. *Photogrammetric Engineering & Remote Sensing*, **66**, pp. 1011–1016.
- PONTIUS, R.G., JR, 2002, Statistical methods to partition effects of quantity and location during comparison of categorical maps at multiple resolutions. *Photogrammetric Engineering & Remote Sensing*, **68**, pp. 1041–1049.
- REED, R.D. and MARKS II, R.J., 1999, *Neural smithing: Supervised learning in feedforward artificial neural networks* (Cambridge, Massachusetts: MIT Press).
- RUMELHART, D., HINTON, G. and WILLIAMS, R., 1986, Learning internal representations by error propagation. In *Parallel distributed processing: explorations in the microstructures of cognition*, edited by D.E. Rumelhart and J.L. McClelland (Cambridge, Massachusetts: MIT Press).
- SOUSA, S., CAEIRO, S. and PAINHO, M., 2002, Assessment of map similarity of categorical maps using Kappa statistics: The case of Sado estuary. In *ESIG 2002* (Tagus Park, Oeiras: Associação de Utilizadores de Informação Geográfica).
- SAWAYA, K., YUAN, F. and BAUER, M., 2001, Monitoring landscape change with Landsat classifications. *ASPRS Proceedings*, St. Louis, Mo. April 21–26, 2001.
- SKAPURA, D., 1996, *Building neural networks* (New York: ACM Press).
- VELDKAMP, A. and LAMBIN, E., 2001, Predicting land use change. *Agriculture, Ecosystems and Environment*, **85**, pp. 1–6.
- WALKER, R., 2003, Evaluating the performance of spatially explicit models. *Photogrammetric Engineering & Remote Sensing*, **69**, pp. 127–1278.