

Developing a multi-network urbanization model: A case study of urban growth in Denver, Colorado

Jida Wang and Giorgos Mountrakis*

*Department of Environmental Resources and Forest Engineering, College of Environmental Science
and Forestry, State University of New York, Syracuse, NY, USA*

(Received 15 September 2008; final version received 12 October 2009)

Urbanization is an important issue concerning diverse scientific and policy communities. Computational models quantifying locations and quantities of urban growth offer numerous environmental and socioeconomic benefits. Traditional urban growth models are based on a single-algorithm fitting procedure and thus restricted on their ability to capture spatial heterogeneity. Accordingly, a GIS-based modeling framework titled multi-network urbanization (MuNU) model is developed that integrates multiple neural networks. The MuNU model enables a filtering approach where input data patterns are automatically reallocated into appropriate neural networks with targeted accuracies. We hypothesize that observations classified by individual neural networks share greater homogeneity, and thus modeling accuracy will increase with the integration of multiple targeted algorithms. Land use and land cover data sets of two time snapshots (1977 and 1997) covering the Denver Metropolitan Area are used for model training and validation. Compared to a single-step algorithm – either a stepwise logistic regression or a single neural network – several improvements are evident in the visual output of the MuNU model. Statistical validations further quantify the superiority of the MuNU model and support our hypothesis of effective incorporation of spatial heterogeneity.

Keywords: urban modeling; land use and land cover change; spatial analysis; neural networks; hybrid classifiers; urban applications

1. Introduction

Urbanization relates to geographic dynamics where rural areas are interfered with and eventually replaced by the expansion of contingent cities or densely populous areas (Alig and Healy 1987, Jacquin *et al.* 2008). There is clear evidence of urbanization worldwide (e.g. United Nations Centre for Human Settlement 1996), further enhanced by recent economic growth in India and China. Urbanization, however, is raising concerns of environmental, ecological, and social nature. Particularly, if land use transitions are ineffectively planned, continuous growth of urbanized areas will perturb and end up breaking the equilibrium of environmentally sensitive areas (Lee *et al.* 1999). Scientific models addressing location and quantity of urban expansion in space and time are essential in providing scientists and policy makers with empirical/statistical support for their decisions toward an environmentally sustainable future.

*Corresponding author. Email: gmountrakis@esf.edu

From the algorithmic development perspective, this article proposes a multi-network urbanization (MuNU) model, motivated by a crucial challenge to land use change models: the ability to capture spatial heterogeneity in geographic data. Spatial heterogeneity describes the variability or uneven distribution of a geographic attribute (Bryk and Raudenbush 1992, Bailey and Gatrell 1995, McDonald and Urban 2006). A traditional modeling structure with a single algorithm fitting the entire study area neglects the various governing rules embedded in local effects and therefore overgeneralizes the patterns of land use change. Keeping a balance between model accuracy and complexity, the MuNU model enables a multi-filtering approach where input data patterns are automatically reallocated into appropriate neural networks with targeted accuracies. The ability to incorporate spatial heterogeneity is validated with our major hypothesis:

The proposed MuNU model using multiple neural networks outperforms in terms of classification accuracy a single-algorithm model – either a back-propagation neural network or a logistic regression model.

We test our model using urban change information on a temporal transition from 1977 to 1997 in the Denver Metropolitan Area (DMA). The remaining of the article is organized into six sections. Following this introduction, Section 2 reviews existing methods for urban land use modeling, such as generalized linear models (GLMs), geographically weighted regression (GWR), cellular automata (CAs), and back-propagation neural networks (BNNs). Section 3 presents the study area, data preprocessing, and techniques of explanatory variable preparation. Section 4 discusses the theoretical underpinnings of the proposed model. Results are visualized and interpreted in Section 5. Models are validated and compared in Section 6 while Section 7 discusses further our research findings.

2. Existing work, challenges, and improvements

The conventional methodology in land use models is the empirical assessment of land use change correlated to a number of explanatory variables, quantitatively measured by transition potential – the likelihood or risk that land would change from one cover type to another (Bonham-Carter *et al.* 1989, Sklar and Costanza 1991, Hall *et al.* 1995, Eastman *et al.* 2005). Parametric statistical methods such as GLMs provide an explicit rule of land use transition universally applicable to the training area. The function of each explanatory variable in land use change is easily interpreted from the parameter estimates (Schneider and Pontius 2001, Pijanowski *et al.* 2002, Gude *et al.* 2006, McDonald and Urban 2006). Cheng and Masser (2003) summarize and compare three common linear models – linear regression, logistic regression, and log-linear regression – used in urban pattern analysis and conclude that logistic regression is generally more advantageous in the type of dependent and explanatory variables, computational method, and normality assumption. However, these conventional regressions assume that urban growth is spatially stationary. Using such global models, the relationships that are intrinsically heterogeneous across space may be inadequately simplified.

An effective remedy for conventional regressions is the GWR. A GWR is essentially a GLM but allows process parameterization that varies over space (Fotheringham *et al.* 2002). The issue of spatial heterogeneity has been considerably addressed by numerous urban models using the idea of GWR (Jensen *et al.* 2005, Dendoncker *et al.* 2007, Luo and Wei 2009, Noresah and Ruslan 2009). However, a major difficulty of GWR is the determination of neighborhoods: the extents to which parameterizations of the central observations are considered correlated to all the local observations within. In most GWR models, the neighborhood of each observation is circumscribed by a kernel with an arbitrary scale or

observation number. There is a lack of an automated training mechanism that empirically determines the optimal kernel scale for each observation. The parameter surfaces produced by a GWR model tend to overfit the study area and have a limited ability to generalize to patterns in different locations or times.

CAs represent another technique for obtaining high-resolution models on urban land use dynamics. An urban CA sees the complex urban system as four components: a cellular space, the state of each cell, the neighborhood of each cell, and a transition rule (Couclelis 1998, Barredo *et al.* 2003). Similar to GWR, an urban CA also introduces the concept of neighborhood for each cell/observation. However, instead of giving parameter estimates that vary spatially, a CA usually applies one global transition rule for the entire cellular space. Once the rule is trained, simulations are repeatedly updated under the CA structure through a sequence of steps until a certain controlling criterion is satisfied (White and Engelen 2000). For each step, the local constraints (the input variables) of the transition rule are simultaneously re-processed on the states or variable values of the neighbor cells estimated from the previous iteration. Spatial heterogeneity is expected to be incorporated by the continuous updates of these spatially constrained inputs (White *et al.* 1997, Li and Yeh 2000, 2002). In this sense, an urban CA is highly sensitive to the transition rule. Simulation uncertainties can be iteratively accumulated and exaggerated through previous estimation errors. The determination of transition rule is therefore crucial to the incorporation of spatial heterogeneity.

Techniques of artificial intelligence (AI) have arisen rapidly as effective tools in calibrating CA transition rules. Numerous algorithms have been used covering fuzzy logic control (Wu 1998), decision trees (Li and Yeh 2004), support vector machines (Yang *et al.* 2008), ant intelligence (Liu *et al.* 2008), and BNNs (Li and Yeh 2002, Yeh and Li 2002, 2003, Almeida *et al.* 2008). In contrast to the applications of conventional statistical methods such as multi-criteria evaluation (Wu and Webster 1998) and logistic regression (Almeida *et al.* 2003), the AI algorithms capture urban land use patterns in a non-parametric approach and incorporate a higher degree of spatial heterogeneity (McDonald and Urban 2006). However, their power of representing spatial heterogeneity is limited by their single-algorithm structure, which targets the entire data into a global AI algorithm.

In our study, the designed MuNU model combines BNNs and a multi-thread structure. We separate the input space into different segments and address each of these sub-problems individually, a concept originally introduced in Mountrakis (2008) and Mountrakis *et al.* (2009). Spatial heterogeneity is hypothesized to be better modeled by this multi-thread structure compared to a single-step BNN or a logistic regression (an example of the conventional GLM). The combined effect of several benefits supports the selection of BNN as the basis to build our multi-thread structure: (1) compared to conventional statistical methods, a BNN has fewer assumptions and is flexible for nonlinear relationships; (2) as a proven AI technique, each BNN by design captures a certain degree of data spatial heterogeneity; (3) a BNN structure is able to output continuous values representing the relative probability of land use transitions. The continuous output values is a crucial characteristic for our methodology to automatically select a BNN given a targeted classification accuracy (refer to Section 4 for details).

The major contribution of the MuNU model targets improvements in spatial heterogeneity incorporation through a multi-thread model based on BNNs. Compared to other alternatives such as GWR, our model releases the constraint of model parameterization from the observation level. Thus it is more efficient in structure and less constrained in space. As a further extension, our model structure could be adopted in future to calibrate transition rules of CA-based urban growth models.

3. Study area and data preprocessing

3.1. Study area and data sources

The DMA is selected as the study site. The DMA (Figure 1) is located in the center of the Front Range Urban Corridor, between the Rocky Mountains to the west and the High Plains to the east. Urban growth from 1977 to 1997 is used in model development and validation. As the 26th most populous city in the United States, Denver is a typical metropolitan area that experienced rapid population growth and land use changes during this period. By 1997, urban land use had taken over 58% of the total study area compared to 48% in 1977 – more than a fifth of the non-urban land in 1977 had been converted to urban areas by 1997.

The rectangle (Figure 1: UTM Zone 13 North with X coordinate range [478,448.5 m, 518,978.5 m] and Y coordinate range [4,389,826.0 m, 4,421,356.0 m]) covers the major part of DMA and is extracted for study. The land use data set uses classified imagery of the northern Colorado for 1977 (source: 1975, 1977–1978 USGS and USDA aerial photography) and 1997 (source: 1996/1997 imagery), generated for the Front Range Infrastructure Resources (FRIR) Project by the US Geological Survey Rocky Mountain Mapping Center (USGS 2003, [http://rockyweb.cr.usgs.gov/fronrange/data sets.htm](http://rockyweb.cr.usgs.gov/fronrange/data%20sets.htm)). The images are classified using a modified Anderson classification scheme including five levels of land use land cover (LULC) categories. The road data set is also used for variable processing. The elevation image is obtained from USGS National Elevation Datasets in the Colorado area (USGS 2010, <http://seamless.usgs.gov/website/seamless/viewer.htm>). All files are rasterized to a spatial resolution of 30×30 m, as this work is focusing on large-scale patterns.

3.2. Variable Manipulations

3.2.1. Response variable

The response variable is set to a binary class expressing growth of urban developed surface between 1977 and 1997 (Figure 1): one (1) represents land use change from non-developed to developed area (i.e. change) and zero (0) represents non-developed

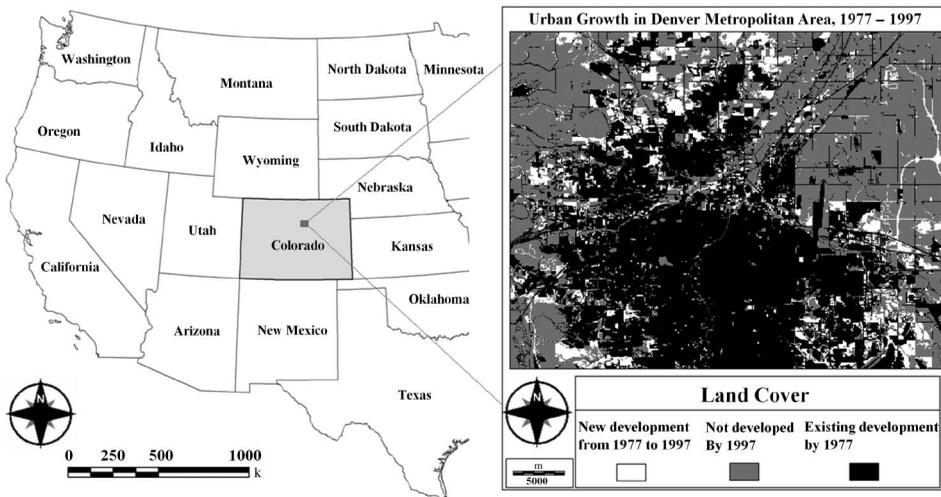


Figure 1. Map of urban growth in Denver Metropolitan Area, 1977–1997.

area that remained non-developed (i.e. no change). The areas that represent existing development by 1977 are excluded from the algorithmic prediction. Seven land use types are defined as urban development: residential areas, commercial/light industries, institutions, communications and utilities, heavy industries, entertainments/recreations, and roads and other transportations. As the objective is to study the pattern and impact of urban growth, the rarely present reverse transition from developed to non-developed areas is not evaluated.

3.2.2. Explanatory variables

Two techniques are used in processing explanatory variables: Euclidean distance to the nearest neighbor (Pijanowski *et al.* 2002) and kernel density filter (Bailey and Gatrell 1995). Both techniques are applied on a binary image corresponding to each variable in 1977 with consideration of practical urban planning. Euclidean distance to the nearest neighbor quantifies the extent potentially influenced by the target features associated with urban expansion, in terms of direct linear accessibility. Kernel density measures the spatial agglomeration convoluted by target neighbors with a weighted contribution expressed by a bell-shaped Equation (1):

$$H = \left(1 - \frac{d^2}{c \cdot r^2}\right)^2 \quad (1)$$

In the above equation r is half the side length of a filtering square window, d is the Euclidean distance from a neighbor (pixel within the square window) to its center pixel, c is an adjustable constant, and H is the weight of the neighbor in the spatial agglomeration. In our calculation, we assign c to be 2.5 (>2) to allow some contribution of the edge pixels.

We examined 18 factors as candidates of explanatory variables (Table 1). Factors X1 through X15 are processed based on the binary images of the target features extracted from the LULC and road data sets. The proximity image of existing urban area in 1977 is filtered using a kernel function to produce X16. Different from X7 to X15, factor X16 incorporates six variables processed at different scales (search radii). The kernel function on the proximity images creates a metric that quantifies the weighted agglomeration of linear accessibility from a center pixel, whereas the multiple filter radii for the kernel function represent the possible variation of local trends at various scales. The factors are concluded with X17 (Elevation) and X18 (Slope).

4. Model development

Moving to our algorithmic development, this section begins with an overview of the general methodology, followed by details on the design structure, information flow, and outputs of our MuNU model.

4.1. Methodology

The fundamental rationale of our method is to embed the spatial heterogeneity into a multi-thread classification framework. By combining several BNN algorithms we strive to improve classification accuracy for their associated data segments. Specifically, we begin with a single BNN on the entire data (all pixels) and evaluate its algorithmic performance.

Table 1. Summary of explanatory variables, in the Denver Metropolitan Area, 1977–1997.

Factor name		Description
Euclidean distance creating proximity images		
X1: Entertainments		Distance to entertainments/recreation
X2: Heavy industries		Distance to heavy industries
X3: Rivers/canals		Distance to rivers/canals
X4: Primary roads		Distance to primary roads
X5: Secondary roads		Distance to secondary roads
X6: Minor roads		Distance to minor roads
Kernel density on binary images	Searching radius (pixel)	
X7: Agricultural business	120	Kernel density of agricultural business
X8: Residential areas	120	Kernel density of residential areas
X9: Urban developments (excluding roads)	120	Kernel density of urban developments
X10: Commercial areas	120	Kernel density of commercial areas
X11: Institutes	120	Kernel density of institutes/schools
X12: Communications/utilities	120	Kernel density of communications/ utilities
X13: Lakes/ponds	120	Kernel density of lands/ponds
X14: Cultivated areas	120	Kernel density of cultivated lands
X15: Natural vegetations	120	Kernel density of natural vegetations
Kernel density on proximity images		
X16: Urban developments (excluding roads)	10, 30, 50, 80, 100, 150	Kernel density of distance to urban developments
Topography		
X17: Elevation (m)		Elevation based on mean sea level
X18: Slope (degree)		Slope

Pixels that are classified with high accuracy are selected and associated with this BNN. Other BNNs are subsequently tested on the remaining data using the same principle. The procedure is repeated until all data are classified. By doing so, we hypothesize that observations in each of these data segments share greater homogeneity and can be modeled more accurately with a collection of targeted algorithms.

We assess our methodology using change or no change data from 1977 to 1997. As our model provides a continuous output in the [0, 1] range, we apply a threshold on each BNN to harden the simulation results and create a binary output. That threshold is inherently capturing an implicit rate of change from 1977 to 1997. Applications of our method in future predictions should either assume the same rate of change or may artificially alter that rate through proper threshold adjustment (e.g. using local land use constraints as a guide).

To formulate our data set we first assess the presence of spatial autocorrelation. A semivariogram is produced within all scales. Because of the large size of our data set we use a 1% subsample of the data set, a subset large enough to create an unbiased generalization of spatial autocorrelation while it is reasonably manageable with current computational power. The semivariogram indicates a spatial autocorrelation up to 15-pixel distances (450 m). Using this information we have produced three data sets from the study area (excluding the existing development in 1977) summarized in Table 2.

Sample I is used to train the MuNU and two benchmark models (a logistic regression and a single BNN; see Section 5 for details). To avoid any modeling bias from spatial autocorrelation, the size of training sample is limited with reference to the range value of the

Table 2. Summary of samples used in model development and validation.

Sample	Method	Size (pixel/percent)	Usage (identical for all three methodologies)
I	Systematic grid sampling (15-pixel intervals)	3286/0.5%	Calibration: training data ensuring lack of spatial autocorrelation
II	Random sampling	146612/20%	Calibration: simulation accuracy assessment
III	Random sampling (excluded in any prior samples)	542088/74%	Validation: implemented for all statistical comparisons

Note: The remaining 5.5% of the total pixels are reserved for future testing of a spatial autoregressive model.

semivariogram (15 pixels). Sample II is used to estimate the ability of generalizing to random data in model calibration and selection. All models are validated on Sample III, which contains pixels not present in any of the calibration samples.

4.2. Design and information flow

The MuNU model is an integration of eight BNNs. The basic structure of each individual BNN is illustrated in Figure 2. The first three BNNs are using 18 explanatory variables: the six proximity variables (X_1 – X_6), nine kernel density variables based on binary images (X_7 – X_{15} with a 120-pixel scale), one kernel density variable based on the distance images (X_{16} with a 100-pixel scale), and the two topographic variables (X_{17} and X_{18}).

Correspondingly, each BNN is structured as 18 input neurons representing the 18 explanatory variables and 2 continuous output neurons: [Neuron₊] representing 1 (transition from non-urban to urban = change) and [Neuron₋] corresponding to 0 (remaining non-urban = no change). We use sigmoidal or linear functions as activation functions for these two output nodes. We denote the continuous outputs of these two nodes as O^+ and O^- , respectively. In the training process, pixels with $O^+ > O^-$ are categorized as change, while all other pixels belong to the no change category. After the third BNN, training of the remaining BNNs follows the same procedure with the addition of five inputs: explanatory variables X_{16} with kernel radius of 10, 30, 50, 80, and 150 pixels are used to account for more information at multiple scales. As the problem complexity increases, variables incorporating multiple scales are expected to improve model accuracy.

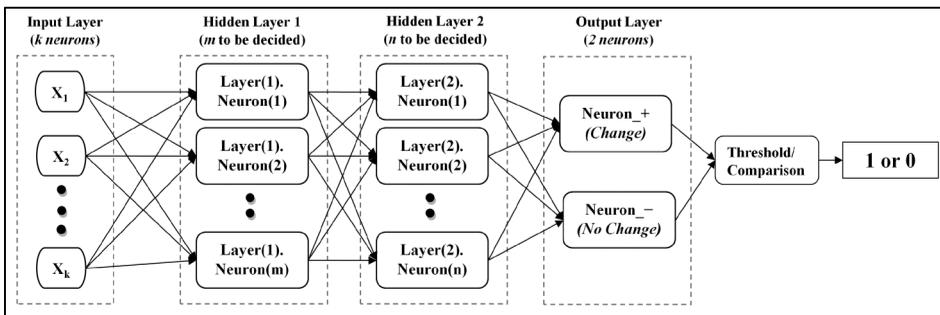


Figure 2. Basic structure of each neural network in the multi-network urbanization (MuNU) model.

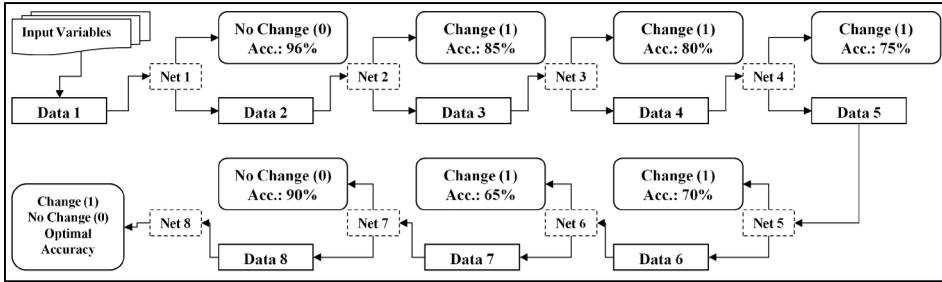


Figure 3. Chain structure of the multi-network urbanization (MuNU) model.

The integration of multiple BNNs is presented in the chain structure of Figure 3. Each of the first seven BNNs is designed to classify either change or no change (Net 1 and Net 7 classify no change, and Net 2 through Net 6 classify change). The basic principle of using multiple BNNs relies on the hypothesis that parts of the problem are easier to address than others. We target specific BNNs to specific parts of the input space by allowing an expert-based selection of optimal algorithms. We start with less complex targets, for example, in our case, with pixels that we are highly confident will not experience urban change (Net 1 in Figure 3). Since 100% correct classification of the entire pixels is not expected, we place an accuracy threshold and identify all pixels classified with that accuracy. For example, in the case of Net 1 and due to the fact that the calibration sample is biased (ratio of no change to change is 3:1), we identified no change class accuracy of 96% as a target goal for the first BNN classifying exclusively no change pixels. The pixels classified by a selected BNN are extracted from the data set and the remaining pixels are fed into the next step. The above process is applied in an iterative way with the accuracy threshold requirements gradually decreasing as problem complexity increases (Figure 3). At the final step (Net 8) and in order to classify all pixels we do not place any accuracy requirements; instead we select the BNN that classifies the remaining change and no change pixels with the highest overall accuracy.

An important component within the MuNU training is to identify the optimal BNN for a given accuracy in each of the steps except Net 8 (Figure 3). We tested 2000 different architectures (i.e. number of hidden nodes and layers) in each of these eight steps to identify each of the eight optimal BNNs. From Net 1 through Net 7, the training requires identification of pixels classified with accuracy higher than a given threshold (e.g. 96% overall accuracy for Net 1, 85% for Net 2, and so on). The accuracy thresholds are selected to progressively decrease and therefore reflect the increasing difficulty of classifying leftover pixels in later BNNs. The actual threshold values in our application are chosen with non-experts in mind and to provide an approximate uniform accuracy sampling (i.e. every 5%). These thresholds are important as their spatial distribution can be visualized (see later in Figure 7). Furthermore, this model implements eight BNNs to exhibit enough spatial accuracy variability while improving overall classification accuracy. In future work we plan to investigate the effects of the number of selected BNNs.

As most of the 2000 candidate architectures would classify some pixels with the predetermined accuracy, we identify as the optimal architecture the one that classifies the highest number of pixels with the given accuracy, in other words the one with the highest absorption rate. The optimal BNN for Net 8 is selected as the one with the highest overall accuracy. The BNN selection criteria, e.g. the absorption rate and the overall accuracy, are calculated from the simulation on a random sample representing 20% of the total pixels (Sample II) (see the pseudo-code in Appendix 1).

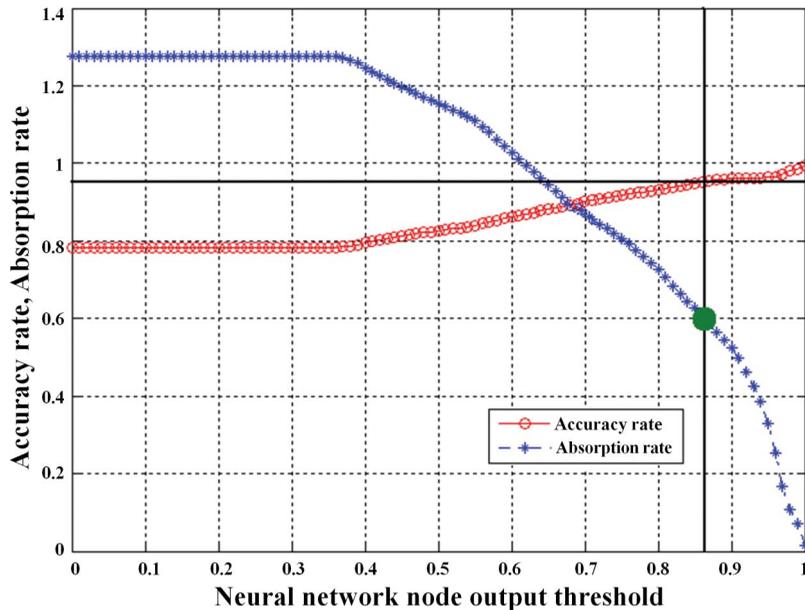


Figure 4. Illustration of accuracy and absorption rate calculation.

Figure 4 illustrates the scheme of quantifying the maximum absorption rate. Calculations of both accuracy and absorption rates are based on the node outputs of each BNN. Depending on the class each BNN is classifying, O^- is used for Net 1 and Net 7 (no change) and O^+ is used for Net 2 to Net 6 (change). In each architecture of Net 1, for example, the corresponding output values are reordered and categorized into numerous thresholds within the $[0, 1]$ output range. The corresponding accuracy rate at each threshold is calculated as the proportion of correct classifications of no change in the subset pixels whose O^- node output is greater than this threshold. The corresponding absorption rate at this threshold is the ratio of the subset size (number of no change classifications with O^- greater than the threshold) to the total number of pixels of no change in the ground truth. Values may reach above 1 (100%) when the classifier absorbs a high number of pixels from the opposing class, but accuracy will suffer.

In most cases, decrease of the threshold (O^-) leads to a lower classification accuracy of no change (0) and thus higher absorption rate (i.e. higher number of classified pixels), as indicated in Figure 4. In Net 1 we assume an accuracy rate of 0.96 (marked by the horizontal line); the subset of pixels with O^- output threshold larger than 0.86 (marked by the vertical line) corresponds to the output threshold that provides the requested classification accuracy of no change while it concurrently classifies the higher amount of pixels (highest absorption rate). The maximum absorption rate satisfying the requested 0.96 accuracy target is identified at ~ 0.60 in this example. The selected BNN for Net 1 is the one with the maximum absorption rate in 2000 BNN candidates repetitively trained and simulated as illustrated. After the optimal BNN is identified, pixels with O^- larger than its hardening threshold (0.86 in this example) are categorized as no change (0) and are extracted from the data set. By doing so, we make a rational assumption that pixels with high O^- values have little chance to coincide with a greater O^+ value (no exceptions are found in Sample II), and thus unlikely to be categorized as change and vice versa. In Net 7 where no changes (0) are classified, the same procedure is followed. In Net 2 through Net 6 where changes (1) are the target of the classifier, thresholds are transformed using BNN output from node O^+ instead of O^- , and the accuracies and absorption

rates are calculated based on classifications of change (1). As mentioned above, the final BNN (Net 8) is selected as the one which outputs the maximum overall accuracy of classifying both change (1) and no change (0) from the 2000 attempts simulated on Sample II without any consideration for absorption rate (i.e. absorption rate is 100% since this is the last step).

All BNNs are trained using the Levenberg–Marquardt back-propagation algorithm. The pseudo-code is provided in Appendix 1 to demonstrate specifics on the architecture, training, and selection of each BNN in the MuNU model.

4.3. Simulation outputs

Given the calibrated structures and input data, the MuNU model provides a direct output containing each of the two neuron nodes O^+ and O^- for each of the eight BNNs for a total of 16 direct outputs. These outputs are further processed to calculate the following three metrics:

- (1) \hat{y} : binary prediction of either change or no change;
- (2) p : probability of change;
- (3) acc: spatial accuracy.

Using the selected absorption thresholds described in the previous section we identify the actual BNN classifying each input vector and convert the continuous node output into a binary prediction \hat{y} . In Net 1 to Net 7, \hat{y} is decided by a predefined threshold. Sequentially, the entire pixels are first simulated by Net 1. Pixels with values of O^- larger than the hardening threshold are categorized as 0 and extracted. The remaining pixels are fed into Net 2. Pixels with values of O^+ larger than the hardening threshold are categorized as 1 and extracted from the remaining pixels. The similar procedure is repeated from Net 3 through Net 7. In Net 8, \hat{y} is calculated by evaluating the strength of node outputs O^+ and O^- . If $O^+ > O^-$ then \hat{y} is set to 1; otherwise, \hat{y} is set to 0.

Calculations of the latter two outputs are based on the simulations on Sample II. In Net 1 through Net 7, the continuous O^+ or O^- outputs are divided into numerous intervals, each treated as a separate bin. In Net 8, where both changes and no changes are classified, the difference between O^+ and O^- is categorized into bins. The probability of change (p) is calculated as the proportion of actual change in the total pixels falling in each bin. Spatial accuracy (acc) is calculated as the proportion of correct classifications in the total pixels falling in each bin independently of the class they belong to.

This multi-step approach not only provides higher overall accuracy but also spatially explicit accuracy metrics. Metrics such as probability of change and spatial accuracy are outputs based on each bin, which is the smallest unit of the spatial segmentation in our algorithm. Spatial error estimation is a highly desired characteristic because large-scale urban growth maps may be expected to act as inputs for further analysis by scientists – e.g. urban modelers, biologists, and hydrologists – who are not familiar with image classification or modeling methods (Mountrakis 2008, Mountrakis *et al.* 2009).

5. Results and visual interpretation

From the urban planning perspective, we first provide a brief statistical discussion on the contribution of each explanatory variable to urban growth. Then we assess our MuNU model through a visual comparison against two traditional methods based on a single algorithm: a stepwise logistic regression and a single BNN (a single-net model). The final structure for each model is followed by a visual interpretation of the results.

5.1. Contributions of explanatory variables

To quantify the empirical contribution of each factor, a univariate analysis of variance (ANOVA) is conducted for each of the 23 explanatory variables based on the calibration sample. The F and p values are output in Table 8 (Appendix 2). Twenty variables are tested with p values smaller than $\alpha = 0.05$, rejecting the null hypothesis that the variable means in either change or no change scenario are equal. Variables such as distance to entertainment (X1), density of residential areas (X8), density of urban development (X9), density of natural vegetations (X15), and density of proximity to urban development (X16 with all 6 radii) indicate F values larger than 150 ($p < 0.0001$), suggesting a greater capability to classifying urban growth. Factors such as distance to rivers/canals (X3) and density of lakes/ponds (X13), on the other hand, indicate minor contribution to urban growth. Compared to elevation, slope plays a much more important role in urban development.

5.2. Identified model structures

The two benchmark models – logistic regression and single BNN – are trained on the same data set (Sample I) and the same 23 explanatory variables used in the MuNU model. To decrease multi-collinearity, the logistic regression employs the first nine principal components of the explanatory variables, which account for 99.58% of the total variance. The outputs from the logistic model are reordered from 1 to 0. To convert these continuous outputs to a binary classification (change or no change), the logistic model is simulated on Sample II and a threshold is determined (0.125) as the value where the volume of pixels with larger outputs is approximately equal to the volume of change in the ground truth. Coefficients of the logistic model are provided in Appendix 2. The positive or negative sign of each coefficient interprets the functionality of each variable, which may either promote or prevent urban growth.

In the case of the single-net model we use the same training data set (Sample I) and the best BNN is selected from 2000 candidates based on the highest overall accuracy tested on Sample II. The hardening of the single BNN is performed by identifying the stronger response between the two output nodes for change and no change. For our MuNU model eight different BNNs were selected and integrated into a seamless model. The internal BNN architecture of each of the eight BNNs is shown in Table 3. The ‘Target class’ column refers to whether that BNN captures change, no change, or both. The ‘Accuracy threshold’ column identifies BNNs classifying portions of the data set to the specified accuracy; value of none suggests full absorption of the data set. In ‘Neurons of hidden layer’ we first report the number of neurons in the first hidden layer followed by the number of neurons in the second hidden layer, with 0 indicating lack of a second hidden layer. The hardening threshold for each BNN is reported in the last column. For comparison purposes we also provide the relevant information from the single-net model. Besides structure and hardening threshold, the algorithm of each BNN also varies with the activation function and the adjusting weight that connect the neurons between contiguous layers; however, due to the voluminous information, we do not detail them in this article.

5.3. Visual outputs and interpretation

For both the logistic and single-net models, three outputs are simulated and compared with our MuNU model: the binary predictions (Figure 5), the probability of change (Figure 6), and the spatial accuracy (Figure 7). For the logistic model, the latter two outputs are

Table 3. MuNU and single-net characteristics.

Network	Target class	Accuracy threshold (%)	Neurons of input layer	Neurons of hidden layer	Neurons of output layer	Hardening threshold
Single	0/1	None	23	15/3	2	$O^+ > O^-$ as 1, others 0
MuNU.Net 1	0	96	18	10/4	2	$O^- > 0.90$
MuNU.Net 2	1	85	18	8/4	2	$O^+ > 0.71$
MuNU.Net 3	1	80	18	15/0	2	$O^+ > 0.78$
MuNU.Net 4	1	75	23	15/2	2	$O^+ > 0.89$
MuNU.Net 5	1	70	23	8/4	2	$O^+ > 0.83$
MuNU.Net 6	1	65	23	9/0	2	$O^+ > 0.83$
MuNU.Net 7	0	90	23	10/4	2	$O^- > 0.85$
MuNU.Net 8	0/1	None	23	15/5	2	$O^+ > O^-$ as 1, others 0

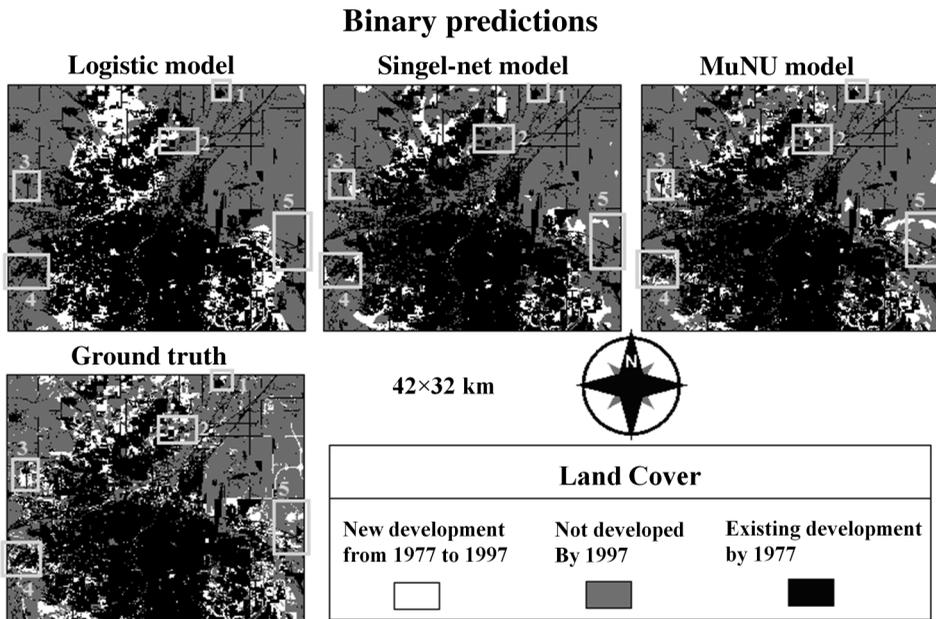


Figure 5. Binary predictions.

calculated against the bins generated on the output values of logistic regression in the range of $[0, 1]$. For the single BNN we follow the same approach used in the MuNU model as demonstrated in Section 4.3. As expected, the single BNN outperforms logistic regression, a result already identified by McDonald and Urban (2006). Their findings indicate that some AI algorithms, such as BNNs and decision trees, are more flexible in adopting data and could establish rules that are more spatially heterogeneous.

Compared to the reference image (Ground Truth in Figure 5), both the logistic and single-net models capture the basic pattern of urban growth in three major clusters: the north, the southwest, and the southeast. However, the logistic model does not capture high details of spatial heterogeneity in urban growth; it is not able to detect numerous small-scaled and disintegrated blocks. In this sense, the single-net model outperforms the logistic one in

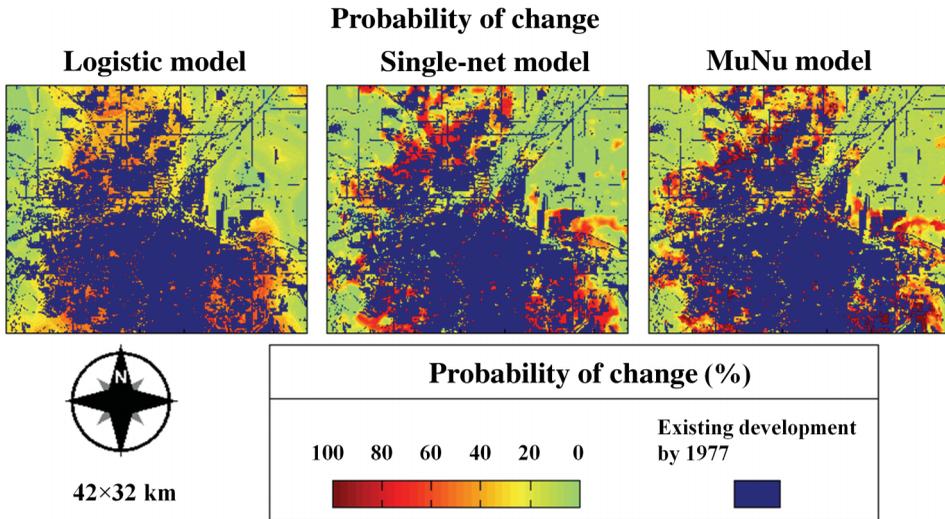


Figure 6. Probability of change.

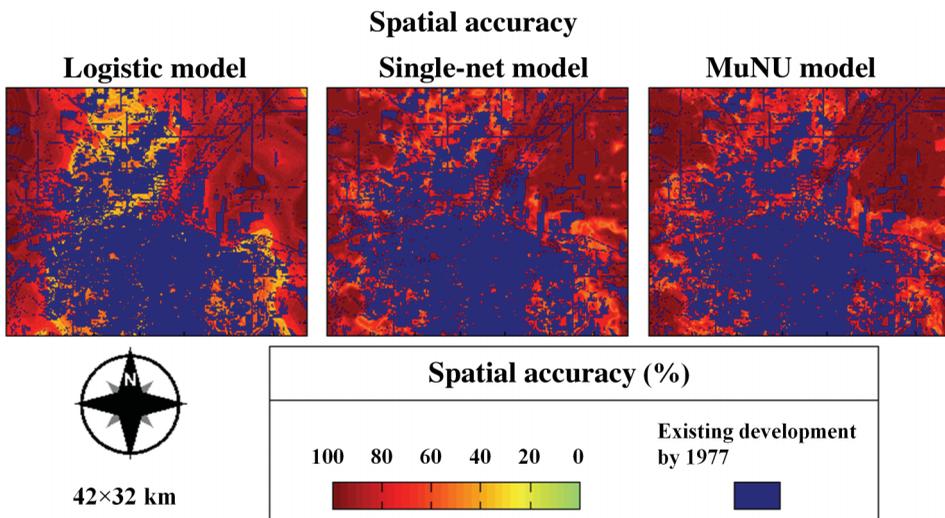


Figure 7. Spatial accuracy.

showing more consistency with the spatial variation of the reference image. Using a multi-thread approach supported by the MuNU model, urban change detection improved further.

Areas bounded by the gray boxes in Figure 5 highlight several local improvements as indicated by greater spatial agreement in the MuNU output. Small and isolated that they are, some of those developments are good indicators of models' capability in incorporating spatial heterogeneity of urbanizations. For example, in the small areas examined in Box 1, the new development northeast to the major existing urban area is completely missed in both logistic and single-net models but mostly captured by the MuNU model. In Box 2, new developments in the ground truth are distributed around existing developments. However, the simulation of the logistic model is skewed toward the western part, and the developments

are almost missed by the single-net model. The MuNU model, again, outperforms the previous ones by capturing the new developments around the central and northern urban areas. Similarly, the new developments in Boxes 3, 4, and 5 are almost missed by the logistic model, showing poor capability of capturing spatial heterogeneity. With the algorithmic improvement from the single-net to the MuNU models, simulations are approaching a higher consistency with the ground truth. All models miss the new developed roads in the middle-eastern section as a new highway was introduced.

The probability images (Figure 6) express the likelihood of a non-urban pixel to switch to urban. Higher probabilities from the logistic model are coarsely located around the major existing urban areas. To serve as a meaningful guideline for urban planners or policy makers, we expect a better quality output providing further spatial details. With the model complexity increasing, the spatial patterns of probabilities are getting more explicit. Moving progressively from the single-net to the MuNU model outputs, the highest and lowest probabilities exist in greater consistency with the ground truth (change and no change, respectively), with the MuNU model providing a more explicit representation.

It is also noticeable in the spatial accuracy images (Figure 7) that there are obvious clusters of lower accuracies in the logistic model. The single-net model significantly improves the results. When the MuNU model is applied, two changes are noticeable: (1) clusters of lower accuracies break into much smaller-scale pieces, justifying the mechanism of multi-step method in representing spatial heterogeneity; and (2) higher accuracies are also getting a stronger presence in the entire study area, indicating an improved overall accuracy in space. Quantitative validation for all the outputs is further conducted in the next section.

6. Model validation

The three models (logistic, single BNN, and MuNU) are statistically compared at both per-pixel and neighborhood levels. We include multiple evaluation methods to assess model performance. Model validation is implemented on the simulations of Sample III (Table 2) – a random sample representing 74% of the total pixels and independent of all the other calibration samples previously used. The validation includes the confusion matrix, Kappa's relative/receiver's operating characteristic (ROC) curve, and multi-scale summary accuracy.

6.1. Pixel-based validation

6.1.1. Confusion matrix

The confusion matrix of the binary prediction (Congalton and Mead 1983) is provided in Table 4 for each model. Three statistics are compared in each confusion matrix: overall accuracy, producer's accuracy, and user's accuracy. The overall accuracy is a general measure for the classification performance: the proportion of the total number of predictions that are correct. The producer's accuracy is the probability that each category in the reference data is correctly classified. In contrast, the user's accuracy indicates the probability that each category resulting from the classifier actually matches the reference data. The result indicates a sequential increase in the overall accuracies: 76.28% for the logistic model, 82.56% for the single BNN, and 83.17% for the MuNU model. The single BNN significantly outperforms the logistic model in most accuracy terms. This result justifies the use of BNN as the basis for building up our multi-step model. Given the major interest in classifying changes, an increase can be noticed in the producer's accuracy for change (1): from 41.22% of the single

Table 4. Confusion matrices of the three models using validation data (Sample III, 542,088 pixels).

Models	Prediction	Reference		Total	User's accuracy (%)
		0: 424412	1: 117979	542088	
Algorithms trained on Sample III – assessing spatial heterogeneity					
Logistic reg.	0	360,231	64,382	424,613	84.84
	1	64,181	53,294	117,475	45.37
Single net	Producer's accuracy	84.88%	45.29%	–	76.28 (overall)
	0	399,058	69,174	468,232	85.23
MuNU	1	25,354	48,502	73,856	65.67
	Producer's accuracy	94.03%	41.22%	–	82.56 (overall)
MuNU	0	397,432	64,233	461,665	86.09
	1	26,980	53,443	80,423	66.45
	Producer's accuracy	93.64%	45.42%	–	83.17 (overall)

BNN to 45.42% of the MuNU model. This indicates that the multi-step approach succeeds in recognizing more urban changes correctly by addressing spatial heterogeneity among the data. The MuNU model also slightly improves the user's accuracy for change (1) from 65.67% in a total of 73,856 pixels predicted by the single-net model to 66.45% in 80,423 pixels. This increase indicates that the MuNU model could provide users (such as policy makers and urban planners) with a more reliable change representation forecasting larger areas of urban changes with higher accuracy.

Another benefit of the MuNU model is that each BNN is targeted with specific accuracy metrics. These metrics are predefined in the training process and tested again on the validation sample (Sample III). The proportions of the validation sample classified by each of the eight BNNs comprised in our MuNU model are provided in Table 5. The reader may notice the small scene contribution of Net 2 up to Net 6. This is due to the following: (1) the urban growth (change) takes a much smaller proportion (22%) of the total validation pixels and (2) the identified BNNs within the MuNU consider a trade-off between the contribution proportion (i.e. percent of the scene they classify) and the prediction accuracy. By using the multi-thread approach, the classifying potential of each data segment identifies the optimal contribution associated with a predefined accuracy and the maximum classification proportion. The overall classification accuracy is based on a weighted average of different BNNs, leading to the summary values for the MuNU model in Table 4.

Table 5. Accuracy summary of each BNN in the MuNU, using validation data (Sample III, 542,088 pixels).

BNNs	Prediction	Reference		Total	Portion (%)	Accuracy (overall 83.17%)	
		0: 424,412	1: 117,979	542,088	100	Target (%)	Actual from validation (%)
Overall							
Net 1	0	211,543	8,743	220,286	40.64	96	96.03
Net 2	1	1,469	8,453	9,922	1.83	85	85.19
Net 3	1	1,664	6,906	8,570	1.58	80	80.58
Net 4	1	3,248	9,480	12,728	2.35	75	74.48
Net 5	1	2,835	6,268	9,103	1.68	70	68.86
Net 6	1	3,248	5,894	9,142	1.69	65	64.47
Net 7	0	53,786	6,200	59,986	11.07	90	89.66
Net 8	0	132,103	49,290	181,393	33.46	–	72.83
	1	14,516	16,442	30,958	5.71	–	53.11

6.1.2. Location and quantity accuracies using kappa statistics

Compared to the accuracies used in a confusion matrix, kappa statistics exclude the correctness due to chance; therefore kappa outputs a metric explicitly quantifying the classification accuracy of a model (Cohen 1960). Four kappa statistics are presented: the traditional kappa (Kstandard), a revised general kappa defined as kappa for no ability (Kno), and two more detailed kappa statistics to distinguish accuracies in quantity and location (Kquantity and Klocation). The Kno statistic is an improved general statistic over Kstandard as it penalizes large quantity errors and rewards further correct location classifications, while Kquantity and Klocation are able to distinguish clearly between quantification error and location error, respectively. Further discussion on the latter three Kappa statistics is provided by Pontius (2000).

Table 6 summarizes the Kappa statistics. The results indicate the progressive improvement from logistic to single-net to MuNU models regarding the Kstandard statistic. Also, kappa for no ability is improved starting with 52.57% for the logistic regression and ending with 66.35% for the MuNU model. In terms of Kstandard, the asymptotic standard error (ASE) and the upper and lower bounds of the 95% confidence interval for each model are provided. The null hypothesis of the provided kappa Z-test is that there is no real agreement between the predictions and the ground truth ($\text{kappa} = 0$) (Cohen 1960, Gwet 2002). Based on the result, all three models output p -values of kappa Z-score much smaller than 0.05, indicating that the null hypothesis is rejected and all predictions statistically agree with the ground truth.

The use of the other two statistics – Klocation and Kquantity – allows us to distribute the Kno results into an assessment for location and quantity accuracy, respectively. Ideally both metrics should be high; depending on the application overall quantity may have priority (e.g. in whole study area comparisons) or location accuracy may be selected (i.e. in smaller scale comparisons). In our case we do not consider the fluctuation of Kquantity as an important concern because values are fairly high (>80%) – a level described as ‘almost perfect’ by Landis and Koch (1977). Regarding the Klocation the proposed MuNU model improves the single BNN by 2.28% ($[57.15\% - 56.15\%]/[100\% - 56.15\%]$) of the remaining misclassifications in terms of location excluding the existing accuracy due to chance. A more detailed look at accuracy at various spatial scales is provided in Section 6.2.

To compare the statistical difference between the proposed MuNU model and the other two benchmark models we conduct Z-tests between Cohen’s kappas (Kstandard), as shown in Table 7. The null hypothesis of the test is that there is no statistical difference between these two Kappas, implying that the two tested models are statistically equivalent in

Table 6. Summary of kappa statistics for the models on validation data (Sample III, 542,088 pixels).

Kappa statistic	Logistic	Single net	MuNU
Kstandard	30.19%	40.72%	44.10%
Kno	52.57%	65.12%	66.35%
Klocation	30.22%	56.15%	57.15%
Kquantity	100.02%	88.94%	90.18%
Kstandard conf. limit			
ASE of Kstandard	0.0015	0.0016	0.0015
Lower conf. limit	29.89%	40.42%	43.80%
Upper conf. limit	30.48%	41.03%	44.40%
Z-test of Kstandard			
Z-score	222.2425	311.8299	333.5285
<p>-Value</p>	<0.0001	<0.0001	<0.0001

Note: Null hypothesis is $\text{kappa} = 0$, $\alpha = 0.05$.

Table 7. Z-test between Cohen's Kappas (Kstandards) using validation data (Sample III, 542,088 pixels).

Z-tests between kappas	Single net vs. logistic	MuNU vs. logistic	MuNU vs. single net
Z-score	48.0127	65.5724	15.4115
p-Value	<0.0001	<0.0001	<0.0001

Note: Null hypothesis is $Kappa_1 = Kappa_2$, $\alpha = 0.05$.

Z score between Kappas is calculated based on Equation (2).

prediction accuracy calculated by the standard Kappa (Cohen 1960). In our case, given a 95% confidence level ($\alpha = 0.05$), if the prediction of one model is statistically different from another model in terms of Kstandard, the p -value should be smaller than 0.05. Table 7 shows that model comparisons are statistically significant with p -values smaller than 0.0001.

$$Z = \frac{K_1 - K_2}{\sqrt{(ASE(K_1))^2 + (ASE(K_2))^2}} \quad (2)$$

where K_1, K_2 represent Kstandard for any two models, and $ASE(K_1), ASE(K_2)$ denote ASEs of any two models.

6.1.3. Probability of change (relative/receiver's operating characteristic)

The relative/receiver's operating characteristic (ROC) curves are used to evaluate the performance of an urban land use model with respect to its accuracy of soft prediction – probability of change (Pontius and Schneider 2001, Pontius and Pacheco 2004). Generally speaking, the ROC curve is the locus of the pair of true positive rate (TPR) and false positive rate (FPR) for each probability of change (Swets 1973, Zweig and Campbell 1993). The higher the curve above the diagonal line formed for $TPR = FPR$, the better the model's accuracy is (Chawla *et al.* 2001). A common summary statistic to describe the ROC curve is area under the curve (AUC). In this case, a measure with perfect predictive power would yield a value of 100% of the total area while one with no power would approach 50%, covering only the area under the diagonal line.

ROC curves and their AUC values for the three models are shown in Figure 8. All curves are above the diagonal, indicating all three models are systematically successful. As indicated by the confusion matrix and Kappa statistics, AUC again justifies the selection of BNN as the basic training method. By addressing the spatial heterogeneity in the MuNU model, there is a noticeable improvement for most levels of cut-off probability (particularly in the upper and lower levels), with AUC increasing from 82.72% to 83.80%.

6.2. Neighborhood-based validation (model applicability at various scales)

The overarching goal of urban modeling is to provide scientific guidance to important policy decisions made by transportation planners, environmentalists, and policy makers. Typical implementations of our model are expected to be at various regional levels – a coarser scale than the 30 m resolution that we are working on. Potential users may define a much larger modeling spatial scale, for example, working at the neighborhood, school district, or suburb scale. To assess model performance at different spatial scales, a metric expressing the summary accuracy in terms of change quantity (Costanza 1989, Pijanowski *et al.* 2002,

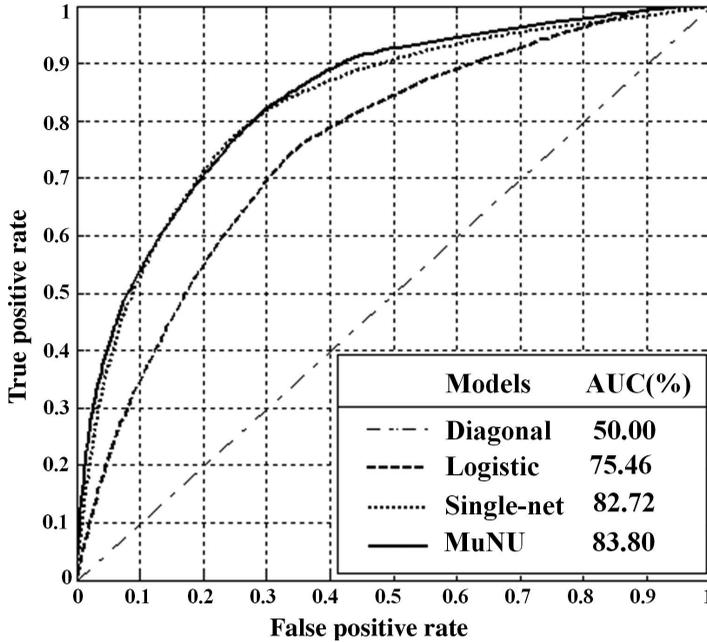


Figure 8. Relative operating characteristic curves for the models.

Pontius *et al.* 2008) is calculated against variable window sizes for the binary outputs (Figure 9). The calculation is defined as

$$\text{Acc}_{i,d} = \frac{\sum_{i=1}^n \left(1 - \left| \frac{m_{i,d} - \hat{m}_{i,d}}{M_{i,d}} \right| \right)}{n} \quad (3)$$

where i denotes a pixel from a total population of n ; d is the diameter (size) of the circular window within which the summary accuracy is examined; $m_{i,d}$ is the actual number of changes in the examined window with a size of d , at the center pixel i ; $\hat{m}_{i,d}$ is the predicted number of changes in this examined window; and $M_{i,d}$ is the number of total valid pixels (excluding the existing development by 1977) in the examined window. The $\text{Acc}_{i,d}$ outputs the average accuracy of change quantity within a window of diameter d .

We implement this variable moving window approach at various scales using all pixels from the validation Sample III (Figure 9). We test multiple window diameters (sizes) ranging from 1 to 135 pixels (30–4050 m). When the window size is 1 pixel, this summary accuracy is identical to the overall accuracy presented earlier in the pixel-based confusion matrix (Table 4). With the increase in window size, the classification accuracy gets less spatially explicit as it only counts the number of correct responses rather than the actual location of these responses. This naturally leads to an increased accuracy for all the models as window size increases. Once again, it is confirmed that the single-BNN model outperforms the logistic model. By addressing spatial heterogeneity, the multi-thread approach significantly improves the single neural BNN by at least $\sim 0.7\%$ at all window sizes, as clearly illustrated in the accuracy difference plot (Figure 9b). The maximum improvement of $\sim 2.8\%$ is achieved at a window size of 1500 m, a relative improvement of larger than 21% if we

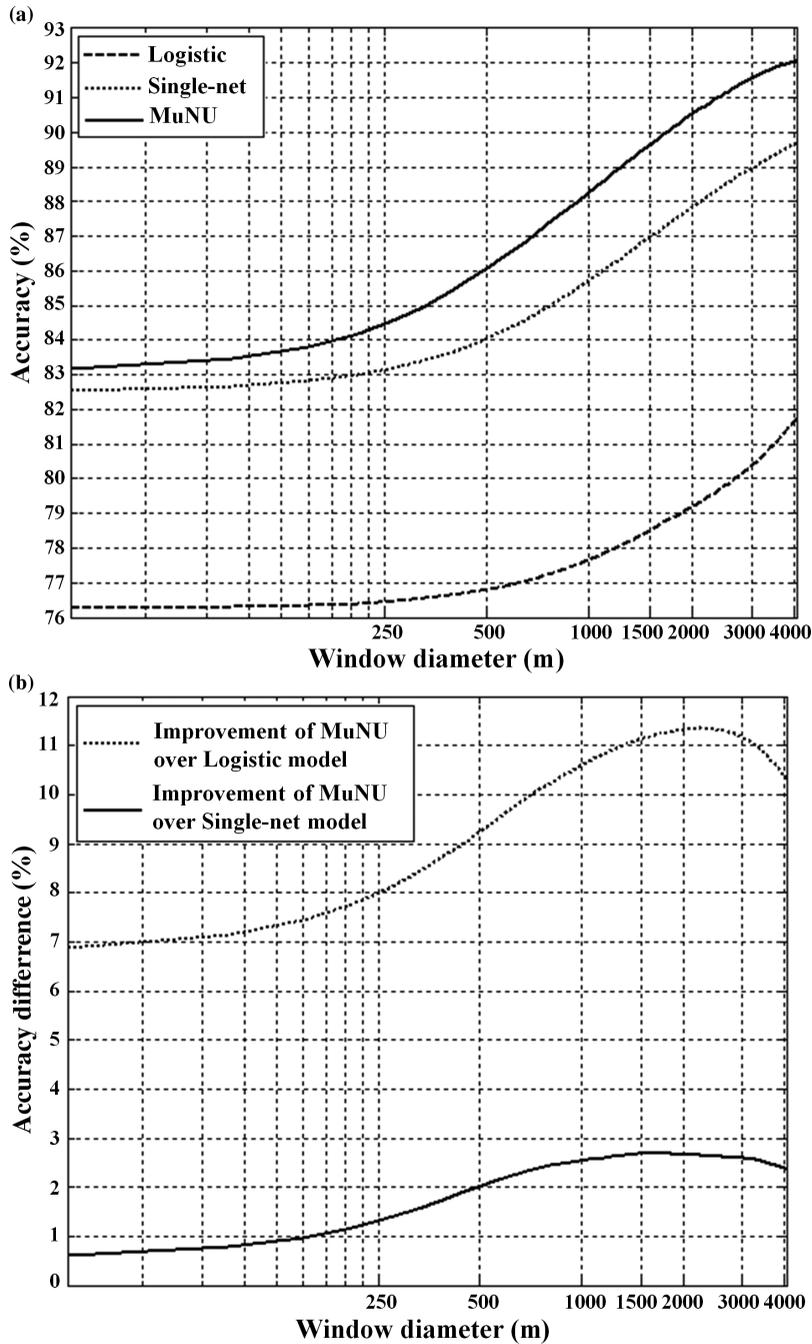


Figure 9. Multi-scale summary accuracies for all models.

consider the accuracy of the single BNN, $(89.7 - 86.9\%)/(100 - 86.9\%)$. This is a significant improvement considering that both methods use identical input data and shows the potential of our methodology.

7. Discussion and concluding remarks

A major motivation behind multi-process algorithms comes from the realization that any near-future increases in computational power would probably result from parallelization rather than raw power increases. Furthermore, recent parallel software developments allow researchers to harvest this significant power increase with minimal programming changes (e.g. using Mathworks Parallel Computing Toolbox). This article acts as an example of how theoretical modeling improvements can complement hardware and software advances.

To match that, we present the MuNU model that takes advantage of multiple neural networks. Even though the eight networks are currently trained in a sequential order, each BNN's internal training can be easily parallelized. That would allow us, for example, to train our eight-BNN MuNU model using a dual-processor quad core machine in approximately the same time it would have taken last year to train a single BNN in a non-parallel fashion. Our theoretical framework provides a more productive alternative to simple training increases of existing single-thread methods that typically reach an accuracy saturation point (e.g. increasing the training of a single BNN from 2000 to 10,000 different architectures would result in minimal improvements in our experiments). Our method is also automated once the initial parameters are set. Therefore, it does not require significant additional user input, beyond the educational component of the modeling process. However, even with current hardware/software improvements, our model is more human and computational resource intensive; therefore, the big question is whether there are substantial benefits associated with the methodology.

To seek an answer we visualize results and perform a detailed validation. From the visualization process the objective conclusion is that MuNU does indeed improve modeling on some local clusters. Our subjective opinion is that this is a significant improvement but the final decision would lie on the actual application task and the user preference. From the statistical validation all per-pixel metrics indicate a small improvement over a single BNN. The degree that these improvements justify the additional complexity of our method would depend on the application in hand. Statistically these improvements are debatable; however, the visualization portion shows that in key areas they are provide an improved spatial distribution of urbanization changes.

On the other hand, the majority of users of urban growth models would be interested in a cumulative accuracy assessment over a neighborhood rather than whether an explicit 30-pixel area has changed. Using Figure 9b as a guide, decision planners have the ability to assess our model performance at different scales and accordingly make a decision as to whether a multi-process model should be implemented. An interesting note from Figure 9b is the actual slope of improvements. We see a slow improvement up to ~ 200 m neighborhoods, followed by a rapid acceleration up to ~ 800 m windows, reaching its peak at ~ 1500 m windows, while benefits slightly decrease at larger scale. Considering the typical urban planning scale of 500–1000 m, absolute improvements lie between 2% and 2.5%. Considering the margin of allowed improvement from a single BNN (i.e. how much better can we do), the relative benefits are between 12% and 17%. This is a highly encouraging result that when combined with the visual inspection of the results allows us to accept our hypothesis, namely that a multi-process model composed of targeted BNNs outperforms a single-process model.

As a final note we should point out two additional significant contributions of this research. First is the incorporation of spatially explicit accuracy metrics accompanying our prediction probabilities. By doing so, non-experts can directly evaluate model performance without reaching an in-depth understanding of the mathematical complexity of our MuNU

model, thus combining advanced modeling capabilities with transparent model evaluation. Second, urban growth modelers often evaluate methods using exclusively per-pixel metrics. There is though a disconnection between such metrics and potential model implementations, where neighborhood metrics may be more important than per-pixel ones. We highly recommend researchers include both metric types in their evaluations.

Acknowledgements

Support for this research was provided by a National Science Foundation award (GRS-0648393) and by a National Aeronautics and Space Administration New Investigator award (NNX 08 AR11G).

References

- Alig, R.J. and Healy, R.G., 1987. Urban and built-up land area changes in the US: an empirical investigation of determinants. *Land Economics*, 63 (3), 215–226.
- Almeida, C.M., *et al.*, 2003. Modeling the urban evolution of land use transitions using cellular automata and logistic regression. In: *Geoscience and remote sensing symposium, proceedings of the 2003 IEEE international conference*, 21–25 July 2003, Toulouse, France. Los Alamitos, CA: IEEE, 3, 1564–1566.
- Almeida, C.M., *et al.*, 2008. Using neural networks and cellular automata for modeling intra-urban land-use dynamics. *International Journal of Geographical Information Science*, 22 (9), 943–963.
- Bailey, T.C. and Gatrell, A.C., 1995. *Interactive spatial data analysis*. Harlow, UK: Prentice Hall.
- Barredo, J.I., *et al.*, 2003. Modeling dynamic spatial processes: simulation of urban future scenarios through cellular automata. *Landscape and Urban Planning*, 64 (3), 145–160.
- Bonham-Carter, G.F., Agterberg, F.P., and Wright, D.F., 1989. Weights of evidence modeling: a new approach to mapping mineral potentials. In: F.P. Agterberg and G.F. Bonham-Carter, eds. *Statistical applications in the earth science, Geological Survey of Canada Paper 89-9*. Ottawa: Geological Survey of Canada, 171–183.
- Bryk, A.S. and Raudenbush, S.W., 1992. *Hierarchical linear models: applications and data analysis method*. Newbury Park, CA: Sage.
- Chawla, S., *et al.*, 2001. Modeling spatial dependencies for mining geospatial data. In: H. Miller and H. Han, eds. *Geographic data mining and knowledge discovery*. London: Taylor & Francis, 131–159.
- Cheng, J. and Masser, I., 2003. Urban growth pattern modeling: a case study of Wuhan City, PR China. *Landscape and Urban Planning*, 62 (4), 199–217.
- Cohen, J., 1960. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20 (1), 37–46.
- Congalton, R.G. and Mead, R.A., 1983. A quantitative method to test for consistency and correctness in photo-interpretation. *Photogrammetric Engineering and Remote Sensing*, 49 (1), 69–74.
- Costanza, R., 1989. Model goodness of fit: a multiple resolution procedure. *Ecological Modelling*, 47 (3–4), 199–215.
- Couclelis, H., 1998. Geocomputation and space. *Environment and Planning. B, Planning and Design*, 25 (anniversary issue), 41–47.
- Dendoncker, N., Rounsevell, M., and Bogaert, P., 2007. Spatial analysis and modelling of land use distributions in Belgium. *Computers, Environment and Urban Systems*, 31 (2), 188–205.
- Eastman, J.R., Solorzano, L.A., and Van Fossen, M.E., 2005. Transition potential modeling for land-cover change. In: D.J. Maguire, M. Batty, and M.F. Goodchild, eds. *GIS, spatial analysis, and modeling*. Redland, CA: ESRI, 357–385.
- Fotheringham, A.S., Brunson, C., and Charlton, M., 2002. *Geographically weighted regression: the analysis of spatially varying relationships*. Chichester, UK: Wiley.
- Gude, J.A., *et al.*, 2006. Prey risk allocation in a grazing ecosystem. *Ecological Applications*, 16 (1), 285–298.
- Gwet, K., 2002. Computing inter-rater reliability with the SAS system. *Statistical methods for inter-rater reliability assessment series*, 3, 1–16.
- Hall, C.A.S., *et al.*, 1995. Modelling spatial and temporal patterns of tropical land use change. *Journal of Biogeography*, 22 (4–5), 753–757.

- Jacquín, A., Misakova, L., and Gay, M., 2008. A hybrid object-based classification approach for mapping urban sprawl in periurban environment. *Landscape and Urban Planning*, 84 (2), 152–165.
- Jensen, R.R., Gatrell, J.D., and McLean, D.D., 2005. *Geo-spatial technologies in urban environments*. Berlin: Springer.
- Landis, J.R. and Koch, G.G., 1977. The measurement of observer agreement for categorical data. *Biometrics*, 33 (1), 159–174.
- Lee, J., et al., 1999. *Development of a community-accessible urban sprawl impact assessment system in northeast Ohio 15-county region for the empact project, Phase One report, literature review for urban growth modeling and environmental impact analysis* [online]. Washington, DC, US Environmental Protection Agency, Grant No. 985989-01-0. Available from: <ftp://gis.geog.kent.edu/EMPACT/review.pdf> [Accessed 5 November 2008].
- Li, X. and Yeh, A.G., 2000. Modelling sustainable urban development by the integration of constrained cellular automata and GIS. *International Journal of Geographical Information Science*, 14 (2), 131–152.
- Li, X. and Yeh, A.G., 2002. Neural-network-based cellular automata for simulating multiple land use changes using GIS. *International Journal of Geographical Information Science*, 16 (4), 323–343.
- Li, X. and Yeh, A.G., 2004. Data mining of cellular automata's transition rule. *International Journal of Geographical Information Science*, 18 (8), 723–744.
- Liu, X., et al., 2008. A bottom-up approach to discover transition rules of cellular automata using ant intelligence. *International Journal of Geographical Information Science*, 22 (11–12), 1247–1269.
- Luo, J. and Wei, Y.H.D., 2009. Modeling spatial variations of urban growth patterns in Chinese cities: the case of Nanjing. *Landscape and Urban Planning*, 91 (2), 51–64.
- McDonald, R.I. and Urban, D.L., 2006. Spatially varying rules of landscape change: lessons from a case study. *Landscape and Urban Planning*, 74 (1), 7–20.
- Mountrakis, G., 2008. Next generation classifiers: focusing on integration frameworks. *Highlight article for Photogrammetric Engineering and Remote Sensing*, 74 (10), 1178–1180.
- Mountrakis, G., et al., 2009. Developing collaborative classifiers using an expert-based model. *Photogrammetric Engineering and Remote Sensing*, 75 (7), 831–844.
- Noresah, M.S. and Ruslan, R., 2009. Modeling urban spatial structure using geographically weighted regression. In: R.S. Anderssen, R.D. Braddock and L.T.H. Newham, eds. *18th World IMACS congress and MODSIM09 international congress on modelling and simulation*, 13–17 July 2009, Cairns, Australia. Canberra, ACT: The Australian National University, 1950–1956.
- Pijanowski, B.C., et al., 2002. Using neural networks and GIS to forecast land use changes: a land transformation model. *Computers, Environment and Urban Systems*, 26 (6), 553–575.
- Pontius, R.G., Jr. 2000. Quantification error versus location error in comparison of categorical maps. *Photogrammetric Engineering and Remote Sensing*, 66 (8), 1101–1016.
- Pontius, R.G., Jr. and Pacheco, P., 2004. Calibration and validation of a model of forest disturbance in the Western Ghats, India 1920–1990. *GeoJournal*, 61 (4), 325–334.
- Pontius, R.G., Jr. and Schneider, L.C., 2001. Land-cover change model validation by an ROC method for the Ipswich Watershed, Massachusetts, USA. *Agriculture, Ecosystems and Environment*, 85 (1–3), 239–248.
- Pontius, R.G., Jr., Thontteh, O., and Chen, H., 2008. Components of information for multiple resolution comparison between maps that share a real variable. *Environmental and Ecological Statistics*, 15 (2), 111–142.
- Schneider, L.C. and Pontius Jr., R.G., 2001. Modeling land-use change in the Ipswich Watershed, Massachusetts, USA. *Agriculture, Ecosystems and Environment*, 85 (1–3), 83–94.
- Sklar, F.H. and Costanza, R., 1991. The development of dynamic spatial models for landscape ecology: a review and prognosis. In: M.G. Turner and R.H. Gardner, eds. *Quantitative methods in landscape ecology: the analysis and interpretation of landscape heterogeneity*. New York: Springer-Verlag, 239–288.
- Swets, J.A., 1973. The relative operating characteristic in psychology. *Science*, 182 (4116), 990–1000.
- United Nations Center for Human Settlement (HABITAT), 1996. *An urbanizing world. Global report on human settlements*. Oxford, UK: Oxford University Press.
- US Geological Survey Rocky Mountain Mapping Center, 2003. *Front Range Infrastructure Resources (FRIR) Project* [online]. Available from: <http://rockyweb.cr.usgs.gov/fronrange/datasets.htm> [Accessed 9 September 2006].
- US Geological Survey Seamless Data Distribution System, 2010. *National elevation datasets* [online]. Available from: <http://seamless.usgs.gov/website/seamless/viewer.htm> [Accessed 15 January 2010].

- White, R. and Engelen, G., 2000. High-resolution integrated modelling of the spatial dynamics of urban and regional systems. *Computers, Environments and Urban Systems*, 24 (5), 383–400.
- White, R., Engelen, G., and Uljee, I., 1997. The use of constrained cellular automata for high-resolution modelling of urban land use dynamics. *Environment and Planning. B, Planning and Design*, 24 (3), 323–343.
- Wu, F., 1998. Simulating urban encroachment on rural land with fuzzy-logic-controlled cellular automata in a geographical information system. *Journal of Environmental Management*, 53 (4), 293–308.
- Wu, F. and Webster, C., 1998. Simulation of land development through the integration of cellular automata and multi-criteria evaluation. *Environment and Planning. B, Planning and Design*, 25 (1), 103–126.
- Yang, Q., Li, X., and Shi, X., 2008. Cellular automata for simulating land use changes based on support vector machines. *Computers and Geosciences*, 34 (6), 592–602.
- Yeh, A.G. and Li, X., 2002. Urban simulation using neural networks and cellular automata for land use planning. In: C. Armenakis and Y.C. Lee, ed. *Geospatial Theory, Processing and Application, Proceedings of ISPRS Technical Commission IV Symposium*, 9–12 July 2002, Ottawa Canada. Vienna, Austria: ISPRS, Vol. XXXIV, Part 4, ISSN 1682–1750.
- Yeh, A.G. and Li, X., 2003. Simulation of development alternatives using neural networks, cellular automata, and GIS for urban planning. *Photogrammetric Engineering and Remote Sensing*, 69 (9), 1043–1052.
- Zweig, M.H. and Campbell, G., 1993. Receiver-operating characteristic (ROC) plots: a fundamental evaluation tool in clinical medicine. *Clinical Chemistry*, 39 (4), 561–577.

Appendix 1. Pseudo-code of the architecture, training, and selection procedures of each BNN

```

% Define parameters. “{}” denotes the variable type and size.
Attempts = 2000;
t = the defined threshold of the classification accuracy, {a scalar};
y = the ground truth binary value for each pixel, {a scalar};
s = the simulated binary value for each pixel, {a scalar};
N = the number of pixels in the sample, {a scalar};
Y = the ground truth binary values (y) for the sample, {a 146612 × 1 vector};
S = the simulated binary values (s) for the sample, {a 146612 × 1 vector};
For each Attempt
  the neuron number of Hidden Layer 1 = integers between 8 and 16;
  the neuron number of Hidden Layer 2 = integers between 0 and 6;
  % Define neural network architecture
  A neural network net is trained using feedforward back-propagation network with a structure of:
  1 input layer:
  the neuron number = the number of input variables;
  2 hidden layers:
  the neuron number of Hidden Layer 1 = the neuron number of hidden layer 1;
  the neuron number of Hidden Layer 2 = the neuron number of hidden layer 2;
  1 output layers:
  The neuron number = 2: [Neuron_+ for 1 and [Neuron_- for 0];
  Transfer function of the input layer = tansig: tangent sigmoid transfer function;
  if the neuron number of Hidden Layer 2 > 0
    Transfer function of Hidden Layer 1 = tansig: tangent sigmoid transfer function;
    Transfer function of Hidden Layer 2 = purelin: linear transfer function;
  else
    Transfer function of the hidden layer = purelin: linear transfer function;
  end if
  Back-propagation network training function = trainlm: Levenberg-Marquardt back-propagation;
  % Neural network training

```

```

The training learning rate = 0.01;
When any one of the following three goals below is met, the training stops.
The training goal = 0.02 in terms of mean square error;
Maximum training epochs = 50;
Maximum training time = 100 seconds
% Neural network simulation
o1 = the simulated output of [Neuron_+], {a scalar for a single vector input};
o2 = the simulated output of [Neuron_-], {a scalar for a single vector input};
if o1 >= o2
    the simulation s = 1;
else
    the simulation s = 0;
end if
% Accuracy calculation and network selection
Error = |Y - S|, {a 146612 × 1 vector};
if the neural network aims to classify 1 (urban growth)
    target = [Neuron_+];
    target_object = 1, {a scalar};
    Num_target = the number of ground-truth 1 (urban growth) in the sample, {a scalar};
    BIN = presenting values between the minimum and maximum of o1, with equal interval of 0.01,
    {an m × 1 vector: m is dependent on the minimum and maximum of o1};
else if the neural network aims to classify 0 (no growth)
    target = [Neuron_-];
    target_object = 0, {a scalar};
    Num_target = the number of ground-truth 0 (no growth) in the sample, {a scalar};
    BIN = a vector, presenting values between the minimum and maximum of o2, with equal interval of
    0.01, {an m × 1 vector: m is dependent on the minimum and maximum of o1};
end if
if the neural network aims to classify both 1 (urban growth) and 0 (no growth)
    K = the number of correct responses in Error (value = 0), {a scalar};
    Accuracy = K/N, {a scalar};
    if Accuracy in this attempt > Accuracy in all the previous attempt
        The best neural network is selected to be the neural network trained in this attempt;
    end if
end if
end if
bin = each element in BIN, {a scalar};
for each bin
    P = the pixels classified as target_object with target values > this bin;
    k = the number of correct response in Error, for pixels P, {a scalar};
    n = the number of pixels of P, {a scalar};
    Accuracy = k/n, {a scalar};
    a = the number of pixels classified as target_object for pixels P, {a scalar};
    Absorption_rate = a/Num_target, {a scalar};
end for
bin_t = the "bin"s whose Accuracy > t, {an m × 1 vector: m is dependent on the number of the
"bin"s whose Accuracy > t};
bin_selected = the bin which has the maximum Absorption_rate in bin_t, {a scalar};
if max(Absorption_rate) in this attempt > max(Absorption_rate) in all the previous attempts
    the best neural network is selected to be the neural network trained in this attempt;
    the bin_selected is used as the threshold: the pixels whose target values > bin_selected will be
    extracted in this neural network and classified as either 1 or 0.
end if
end for

```

Appendix 2. Statistics of the logistic model

The function of the logistic model is expressed in Equation (4):

Table 8. Coefficients of explanatory variables.

Factor name	Coefficients (1×10^{-5})	ANOVA		
		F-value	p-Value	
Euclidean distance creating proximity images				
X1: Entertainments	24.43	213.26	<0.0001	
X2: Heavy industries	-12.88	9.85	<0.0017	
X3: Rivers/canals	-5.54	2.68	<0.1019	
X4: Primary roads	10.19	62.23	<0.0001	
X5: Secondary roads	7.50	20.63	<0.0001	
X6: Minor roads	-0.15	31.15	<0.0001	
Kernel density on binary images	Searching radius (pixel)			
X7: Agricultural business	120	0.18	18.61	<0.0001
X8: Residential areas	120	-6.68	372.53	<0.0001
X9: Urban developments (excluding roads)	120	1.52	314.38	<0.0001
X10: Commercial areas	120	2.35	73.40	<0.0001
X11: Institutes	120	3.79	56.30	<0.0001
X12: Communications/utilities	120	-0.64	54.73	<0.0001
X13: Lakes/ponds	120	-18.60	0.59	0.4414
X14: Cultivated areas	120	7.34	9.63	0.0019
X15: Natural vegetations	120	12.54	159.76	<0.0001
Kernel density on proximity images				
X16: Urban developments (excluding roads)	10	2.13	196.63	<0.0001
	30	1.88	174.39	<0.0001
	50	1.84	211.40	<0.0001
	80	1.86	230.63	<0.0001
	100	1.93	227.19	<0.0001
	150	2.07	219.88	<0.0001
Topography				
X17: Elevation (m)		-0.03	0.15	0.6948
X18: Slope (degree)		-0.003	29.38	<0.0001
Constant		1.57×10^5		

$$p = \frac{\exp(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \dots + \beta_{23} X_{23})}{1 + \exp(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \dots + \beta_{23} X_{23})} \quad (4)$$

where p is the probability of urban growth, β_0 a constant to be estimated, and β_i the coefficient estimated for variable X_i .

The logistic model is fit by the first nine principal analysis components (PCs) using the stepwise method with entry/stay level = 0.01 for each PC. Coefficients of the explanatory variables are converted from the PC coefficient estimates and their eigenvalues. Table 8 summarizes the estimated coefficient and ANOVA statistics for each variable. The positive or negative signs of each coefficient can be used to interpret the contribution effects of each explanatory variable, i.e. either promoting or preventing urban growth. We should note that for the neural network inputs' selection, ANOVA results were not taken into account because neural networks have the internal ability to discriminate patterns.