Environmental Modelling and Software 111 (2019) 182–191

Contents lists available at ScienceDirect

Environmental Modelling & Software

journal homepage: www.elsevier.com/locate/envsoft

The land transformation model-cluster framework: Applying *k*-means and the Spark computing environment for large scale land change analytics

Hichem Omrani[a,*], Benoit Parmentier[b,c,d], Marco Helbich[e], Bryan Pijanowski[f]

[a] Urban Development and Mobility Department, Luxembourg Institute of Socio-Economic Research (LISER), Luxembourg
[b] University of Mary Washington, Department of Geography, Virginia
[c] University of Maine, Mitchell Center for Sustainability Solutions, 5710 Norman Smith Hall, Orono, ME 04469-5710, USA
[d] University of Maryland, National Socio-Environmental Synthesis Center, USA
[e] Department of Human Geography and Spatial Planning, Faculty of Geosciences, Utrecht University, The Netherlands
[f] Department of Forestry and Natural Resources, Purdue University, 195 Marsteller St., West Lafayette, IN 47907, USA

ARTICLE INFO

Keywords:
Clustering
Parallel processing
Spark environment
Land use change

ABSTRACT

This study introduces a novel framework for land change simulation that combines the traditional Land Transformation Model (LTM) with data clustering tools for the purposes of conducting land change simulations of large areas (e.g., continental scale) and over multiple time steps. This framework, called "LTM-cluster", subsets massive land use datasets which are presented to the artificial neural network-based LTM. LTM-cluster uses the *k*-means clustering algorithm implemented within the Spark high-performance compute environment. To illustrate the framework, we use three case studies in the United States which vary in simulation extents, cell size, time intervals, number of inputs, and quantity of urban change. Findings indicate consistent and substantial improvements in accuracy performance for all three case studies compared to the traditional LTM model implemented without input clustering. Specifically, the percent correct match, the area under the operating characteristics curve, and the error rate improved on average of 9%, 11%, and 4%. These results confirm that LTM-cluster has high reliability when handling large datasets. Future studies should expand on the framework by exploring other clustering methods and algorithms.

1. Introduction

Land change science is now challenged with a massive quantity of (freely available) high resolution land cover data that is generated by a multitude of satellite and airborne platforms - many with continental or global coverage. Advancements in computational methods (Denning and Lewis, 2017) have facilitated land change modeling toward larger spatial extents and longer time frames (e.g., Pijanowski et al., 2014; Tayyebi et al., 2013). Numerous models have been developed to simulate land change at a variety of extents (Liu and Phinn, 2003; Yang et al., 2008; Omrani et al., 2015; Shafizadeh-Moghadam et al., 2015; Basse et al., 2016; Azari et al. 2016; Shafizadeh-Moghadam et al., 2017a; b; c). Among these, machine learning-based (ML) models (Pijanowski et al., 2014) have proven their potential in quantifying complex relationships and interactions among drivers while improving land change forecasts (Shafizadeh-Moghadam et al., 2017a; b; Tayyebi et al., 2014b; Omrani et al., 2017a,b). Model fitting over large areas has commonly relied on smaller training samples. Besides, dividing the area into multiple sub-regions and simulating these separately is also common (e.g., see Pijanowski et al., 2014). We propose an alternative method that relies on data partitioning algorithms.

The aim of this research is to introduce a novel data-driven framework, called "LTM-cluster", to address the aforementioned modeling challenges by splitting the input data into clusters using a partitioning algorithm prior to machine learning which we implement in the big data high performance computing (HPC) environment called Spark. We address the following research questions:

1. Does clustering prior to learning scheme coupled with the LTM improve the model's goodness-of-fit during calibration (training) and validation (testing)?
2. Do these techniques work equally well in areas that differ in spatial

Abbreviations: LTM, land transformation model; LTM-cluster, land transformation model-cluster framework; HPC, high-performance computing; ML, machine learning; ANN, artificial neural network; LCS, land change science; PCM, percent correct match; TOC, total operating characteristic; AUC, area under the curve; ER, error rate; MSE, mean square error

* Corresponding author.
E-mail address: hichem.omrani@liser.lu (H. Omrani).

https://doi.org/10.1016/j.envsoft.2018.10.004
Received 5 December 2017; Received in revised form 8 October 2018; Accepted 8 October 2018
Available online 10 October 2018
1364-8152/

extent, cell size, and quantity of land change?
3. What is the gain in computational efficiency of porting traditional ML algorithms to Spark?

To address these questions, we compare the performance of the traditional artificial neural network-based (ANN) Land Transformation Model (LTM; Pijanowski et al., 2014) and the proposed LTM-cluster framework using Spark enabled *k*-means clustering algorithms. Our research builds upon a few studies that utilize data clustering prior to a learning scheme as an alternative for handling large datasets (Ayyadurai and Jayanthi, 2012). To ensure that the approach has broad applicability, we tested our scheme on three diverse land use change datasets from the USA. The study sites vary in size, number of inputs, cell resolution, time interval, as well as in number of observations and quantity of change (i.e., urban gain).

2. Literature review

2.1. Generating a scalable and automated framework for large extent dataset using clustering

Scalability and automation are important aspects when modeling land change. Clustering data prior to model fitting can support both aspects. The objective of data clustering is to determine groupings of data values (Jain et al., 1999; Berkhin, 2006; Jain, 2010; Han et al., 2011). A clustering algorithm divides *n* number of objects into *k* groups based on some metric of similarity. Similarity metrics can be generated from one or many variables (i.e., dimensions or alternatively a collection of patterns). An ideal cluster places a portion of the *n* data into a compact group isolated from other groups. Clustering is often considered as an unsupervised task because no training with specific labels is provided. The process of clustering involves four steps (Jain et al., 1999): feature extraction, pattern proximity determination (i.e., establishing distance values between pairs), clustering (i.e., grouping), and abstraction (e.g., selecting from the clusters a small subsample statistically representing the complete dataset). A variety of clustering methods have been introduced, among which some focus on (1) centroid models (e.g., *k*-means, *k*-medians, *k*-medoids), (2) connectivity models which define distances between objects (often called hierarchical which can be top-down or bottom up), and (3) data distribution models (i.e., those that are grouped according to statistical distributions). We use the *k*-means clustering technique because of its ease of use and well-known performance (Kanungo et al., 2002).

2.2. Leveraging ML algorithms in HPC with Spark

For larger datasets, a variety of technologies are available to ensure that computation, storage, and data workflow are optimized and that scalability can be achieved for potentially larger scale modeling. These computational scaling technologies include the operationalization of the MapReduce concept (Bello-Orgaz et al., 2016), which partitions data into units for parallel processing. Processed units are then mapped back into the larger dataset for further analyses. The Apache Hadoop computation platform is designed to utilize the MapReduce concept by providing a HPC environment including massive storage, a distributed file system, and advanced processing power. Spark is an open-source implementation of an HPC framework that is optimized for the use of R, Python, and Java when big data analysis is required (Hashem et al., 2015). It uses a more complex distributed clustering model (i.e., not the dual MapReduce model) where data are loaded once into memory and numerous operations can be performed on it. Spark is especially proficient at executing ML algorithms that require iterative learning (Xin et al., 2013). Given these properties, we leverage the Spark framework capabilities to implement the LTM-cluster framework with the *k*-means clustering and the ANN-based LTM.

2.3. Extending LTM model with Spark

Several supervised ML methods are frequently used in land change modeling. Common ones are Support Vector Machines (SVM), Random Forest (RF), and Artificial Neural Networks (ANN). SVM and RF are computationally expensive and are not well suited for simulations with large datasets (Tuzel et al., 2007) in contrast to ANNs that are more efficient (Pijanowski et al., 2014; Basse et al., 2014). ANNs search for patterns in data iteratively using learning algorithms that mimic the way that neurons parallel process information in the mammalian brain (Zhang et al., 1998). LTM uses a form of ANN to simulate land use change (Pijanowski et al., 2014; Tayyebi et al., 2014b). LTM has been widely applied and has displayed high accuracy (Pijanowski et al., 2005, 2006; Tang et al., 2005; Olson et al., 2008; Wiley et al., 2010; Ray and Pijanowski, 2010; Ray et al., 2010; Pijanowski and Robinson, 2011; Pijanowski et al., 2011; Moore et al., 2012; Tayyebi et al., 2013; Tayyebi et al., 2014a, b; Omrani et al., 2017a,b; Shafizadeh-Moghadam et al., 2017a).

The literature reports that the multi-layer perceptron, a form of an ANN, outperforms traditional statistical models (e.g., logistic and multinomial regression, regression and classification trees (Feng et al., 2016; Omrani, 2015; Charif et al., 2017; Omrani et al., 2017a,b; Shafizadeh-Moghadam et al., 2017b)), support vector machines, and random forests (Shafizadeh-Moghadam et al., 2017c). However, with the increasing availability of data, new model applications are appearing along with new challenges, especially in the area of model calibration.

A common practice in land change model calibration with large datasets is to use only a small portion of input data (e.g., 5% or 10%) for model building (e.g., Pijanowski et al., 2014; Basse et al., 2014, 2016) or even lower (e.g., Mustafa et al., 2018). Approaches to select input data include random sampling (Mustafa et al., 2018), stratified random sampling (Omrani, 2015Omrani et al., 2017; Basse et al., 2014, 2016; Liu and Feng, 2016; Shafizadeh-Moghadam et al., 2017b; c; Charif et al., 2017) etc. However, these sampling strategies introduce a bias as the performance of ML models depends on the distribution of both the input (e.g., distance to the nearest road) and the output maps (i.e., a map of land change developed from calculating the map difference). Ideally, model building and calibration should include the entire range of input data but existing sampling methods discard a lot of information, leading to lower accuracy in predictions (Pijanowski et al., 2014). Another shortcoming of data sampling is when fitted models are used to forecast future land change scenarios. Values beyond the original input data used for calibration become more prevalent (e.g., development has to proceed further from roads as most land for development proximate to roads has already been converted) which can induce forecasting errors (Basse et al., 2016). In addition, if land change modeling is implemented with increasing spatial extents, longer time frames, and greater number of spatial inputs, a scalable computing framework (cf. Tayyebi et al., 2016; Pijanowski et al., 2014) is needed to deal with these calibration challenges.

2.4. Data and study area

We used land use datasets from three case studies: 1) Muskegon County, Michigan; 2) Lynnfield, Massachusetts-Boston and; 3) Southeastern Wisconsin (Table 1) (hereafter as Muskegon, Boston, and SEWI). The Muskegon case study is located in the west central lower peninsula of Michigan, United States. The region is currently dominated by forest in the northeast, agriculture in the center, and urban areas in the southwest. It covers 1,292 km^2 and urban nearly doubled in area between 1978 and 1998 time. The Boston case study consists of the town of Lynnfield which is located in the northeastern Massachusetts. In 1971, Lynnfield was composed of dense and sparse urban development (36% of the town), mixed deciduous and coniferous forest (43%), wetlands (6%), and open land (12%). By 1999, Lynnfield was composed

Table 1
Size of datasets and cell resolution in three diverse case studies from the USA.

Study area	Quantity of urban-gain	Time interval	Quantity of non-urban	Total cells	Number of inputs	Cell size
Muskegon	6,004 (5%)	1978–1998	110,966	117,012	6	100 m
Boston	629,764 (16%)	1971–1999	3,149,635	3,779,399	8	2 m
SEWI	491,031 (8%)	1990–2000	5,377,707	5,868,738	16	30 m

of dense and sparse urban development (45%), mixed deciduous and coniferous forest (36%), wetlands (6%), and open land (10%). The SEWI case study is situated in the state of Wisconsin in the north-central part of the United States. SEWI comprises seven counties: Kenosha, Milwaukee, Ozaukee, Racine, Walworth, Washington, and Waukesha Counties (Pijanowski et al., 2006; Pijanowski and Robinson, 2011). SEWI is currently dominated by agriculture, urban, and forest, which accounted for more than 86% of the landscape in 2011 (47%, 27%, and 12%). Between 2001 and 2011, the percentage of urban areas increased from 24% to 27%, whereas agriculture and forest decreased by 2% and 0.3%, respectively. More than 60% of lost agriculture contributed to urban gain during the 10-year period. SEWI has undergone remarkable urbanization between 2001 and 2011. The urban expansion rate was 10% and densification rate was 1.5% for 2001–2011. We summarized the characteristic of each case studies in Table 1 and additional information can be found in the literature (Blanchard et al., 2015; Tayyebi et al., 2014b).

For each dataset, we determined the difference between urban-gain and non-urban persistence between two time periods (Fig. 1). We excluded the urban class in the initial time because it is impossible for this urban class to have any urban-gain or non-urban persistence across two time points. Furthermore, a set of variables was defined for each cell serving as driving factors (Table 2, Appendix). Two models (i.e., LTM and LTM-cluster) were developed using six variables in 1978 for Muskegon, eight variables in 1998 for Boston, and sixteen variables in 1990 for SEWI, as inputs and urban change maps between two time periods (1978–1998 in Muskegon, 1971–1999 in Boston, 1990–2000 in SEWI) as outputs. The cells of land use have a spatial resolution of 100, 2, and 30 m in Muskegon, Boston, and SEWI, respectively. Our repository provides the three datasets and related *R* code (Omrani et al., 2018).

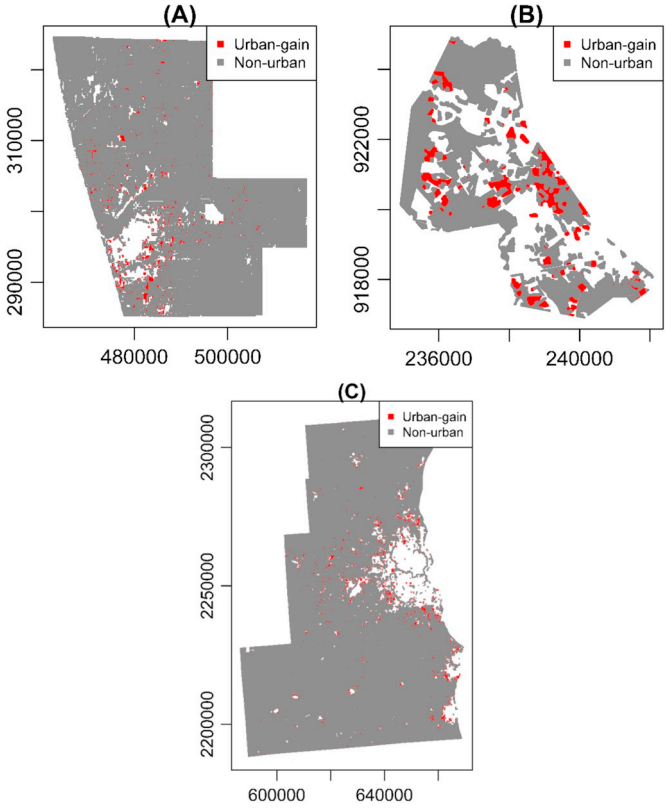

Fig. 1. Urban-gain and non-urban persistence for (A) Muskegon County, (B) Boston, and SEWI (C) during 1978–1998, 1971–1999, and 1990–2000.

3. Methods

The LTM-cluster modeling framework is composed of three phases: 1) generating **clustering** using *k*-means with Spark; 2) LTM **modeling** using HPC; and 3) model accuracy **assessment**. Fig. 2 summarizes the workflow.

The approach is implemented in the *R* programming language (R-3.5.1; Team, 2013) within the Spark environment (Zaharia et al., 2010). To execute LTM-cluster, we used parallel processing including the following steps (Fig. 3):

- Split the dataset S into learning (L) and testing (T) subsets using stratified random sampling. The L set is used for model calibration and the T set is used for assessing its performance (see Section 3.3). We selected 70% of the data to be used for training and the remaining 30% to assess model accuracy (i.e., testing).
- Divide L into *k* clusters (L_k) based on an advanced version of *k*-means algorithm.
- Divide T into *k* clusters (T_k) by assigning cells from T to the closest centroid of L_k.
- Divide S into *k* clusters (S_k) by assigning cells from *S* to the closest centroid of L_k.
- Calibrate and validate LTM-cluster based on L_k and T_k clusters, respectively. This is done by running LTM model for each cluster.
- Simulate urban change for each S_k based on LTM-cluster that is calibrated on L_k.

3.1. Clustering using k-means with Spark

Due to its good performance, the *k*-means algorithm is chosen (Jain, 2010; Murray et al., 2017) for the clustering prior to the learning scheme. The number of clusters is determined by maximizing the average silhouette coefficient (Chiang and Mirkin, 2010) instead of a trial-and-error learning approach (e.g., Basse et al., 2014). The silhouette coefficient is the average distance between all points in a cluster, compared with the average distance between a point and its distance from the nearest cluster. The *k*-means algorithm randomly selects *k* observations as centers of groups and then calculates the Euclidean distances in the feature space between each observation and the centers of all clusters. Next, it assigns to each observation a group, *k*, based on the nearest (e.g., shortest Euclidean distance) cluster center. Centers are iteratively updated until a number of groups is reached and all observations are assigned to a unique cluster.

Because *k*-means cannot handle large datasets due to an excess of computation, we employed Spark-based *k*-means - an advanced version of the standard *k*-means algorithm - to address the computational burden. Spark-based *k*-means performs parallel calculations of distances across all data pairs (Gopalani and Arora, 2015; Meng et al.,

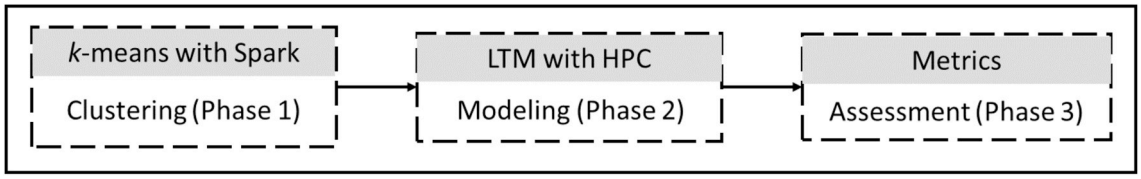

Fig. 2. Main components of the LTM-cluster model.

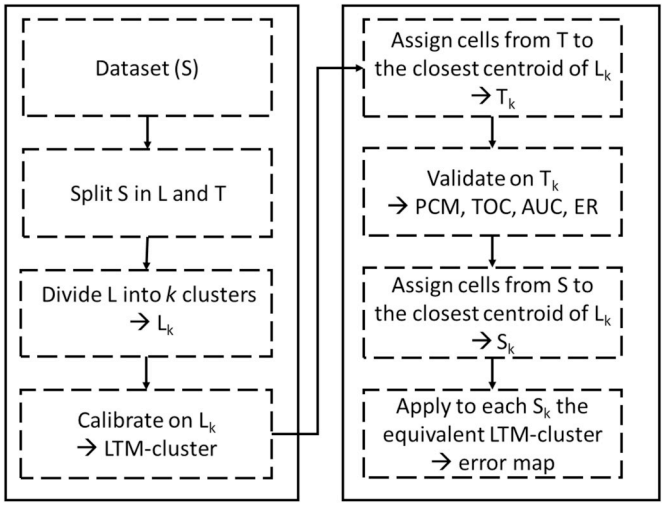

Fig. 3. Conceptual framework (S is the entire dataset; L and T are learning (70%) and testing (30%) subsets generated from S using a random stratified sampling; L_k are clusters resulting from the Spark based *k*-means algorithm; the metrics PCM, TOC, AUC and ER are percent correct match, total operating characteristic curve, area under the *TOC* and error rate (details of these metrics are given in subsection 3.3). T_k and S_k are clustered subsets in which cells are assigned to the closest centroid of L_k. S_k is used with the corresponding LTM-cluster to simulate urban change for the sample S_k.).

2016) while requiring one parameter *k* determined through the average silhouette coefficient to segments the data into *k* clusters. Fig. 3 shows the clustering prior to learning scheme using an advanced *k*-means approach that ultimately minimizes the challenges of a large dataset while improving the LTM's performance. Due to the cutting-edge Spark-based programming techniques, this approach is suitable for exploring large datasets that cannot be stored in a computer's main memory. Studies highlight the fast computation of Spark-based *k*-means compared to alternatives (e.g., basic *k*-means algorithm; Singh and Reddy, 2015; Zaharia et al., 2010).

We also computed the average silhouette value for each value of *k* and for each region in the *k*-means clustering routine. A silhouette value (Rousseeuw, 1987) measures cohesion of each value to its own cluster compared to another cluster. Silhouette values range from +1 (perfect match to its own cluster) and −1 (fits better in another cluster); a silhouette value of 0 means that the value is located, in multi-dimensional space and using a standardized Euclidean distance, precisely between two clusters. Finally, to determine how the best value of *k* varies spatially, we saved the *k* for the largest silhouette coefficient average for each cell and then mapped that for each study simulation area.

3.2. Land Transformation Model

The Land Transformation Model (LTM) uses ANN to derive relationships between land change and multiple explanatory variables (Pijanowski et al., 2014). LTM uses a multi-layer perceptron with one hidden layer with a back propagation training method minimizing the mean square error (MSE). The hidden layer is composed of a set of hidden units called neurons. Hidden layers in ANN allow for the modeling of any nonlinear function (Schmidhuber, 2015) and can approximate the relationship between influential factors (i.e., predictor variables) and outcomes (e.g., a change/no change categorical variable). Several studies have shown that LTM performs well in forecasting an outcome variable (Pijanowski et al., 2005; Omrani et al., 2013, 2015). More formally, an outcome variable *y* contains categorical values in a specified set (*1,2, …,C*). The variable *y* is expressed as a function of the input $x=(x_1,x_2, \ldots,x_q)$ and described by the following formula:

$$P(y_k|)x = \Psi\left\{\sum_{j=1}^{p} v_{jk}\Phi\left(\sum_{i=1}^{q} w_{ij}x_i + w_{0j}\right) + v_{0k}\right\}, \quad (1)$$

where ω_{ij} and v_{jk} are weights assigned to the connections between the input layer and the hidden layer, and between the hidden layer and the output layer, respectively; ω_{0j} and v_{0k} are biases or threshold values in the activation of a unit. Φ is an activation function, applied to the weighted sum of the output of the preceding layer (i.e., the input layer). Ψ is an activation function applied to each output unit, to the weighted sum of the activations of the hidden layer.

An ANN generates outputs contained within the [0, 1] interval but not equal to 0 or 1. A suitable choice for Ψ (e.g., a softmax or a sigmoid function) maps the real axis $(-\infty, +\infty)$ to the interval [0, 1]. ANN's output is a composition that correspond to a set of values summing to unity. Land use conversion are determined using the maximum probability strategy, which assigns each observation to a unique class of land use from the ANN's outputs (Pijanowski et al., 2014; Shafizadeh-Moghadam et al., 2017c). Detailed descriptions of the LTM are available in Pijanowski et al. (2014).

3.3. Model calibration metrics

To validate calibration performance from the LTM and LTM-cluster models, we measure the model's accuracy with four goodness-of-fit metrics. First, we created confusion matrices (Table 3) between the observed and predicted land uses and quantified the location errors (Pontius et al., 2008). We generated error maps that illustrates configuration and location of all errors (i.e., misses, false alarm) and correctly simulated values (i.e., hits and correct rejections). Second, we used Percent Correct Match (PCM, Pijanowski et al., 2014), the Total Operating Characteristic curve (TOC, Pontius and Si., 2014), the area under the curve (AUC, Pontius and Batchu, 2003), and the error rate (ER, Bradley et al., 2016) to quantify the mismatch between the observed and simulated values. PCM specifies the percentage of cells for the land use class correctly classified by a model (true positives*100 / #urban change cells). TOC overcomes the limitations of the relative operating characteristic (ROC), given that the ROC fails in cases where some types of error are more important than others (Dodd and Pepe, 2003). TOC evaluates the performance of non-binary model output by varying the decision threshold between 0 and 1 (Pontius and Si, 2014). From the TOC, we extract the AUC for each case study. Finally, we use the ER metric to measure the incorrect inclusion and incorrect omission of a class (Table 3, Appendix). Larger values of PCM and AUC, as well as smaller value for ER, correspond to better model goodness-of-fit.

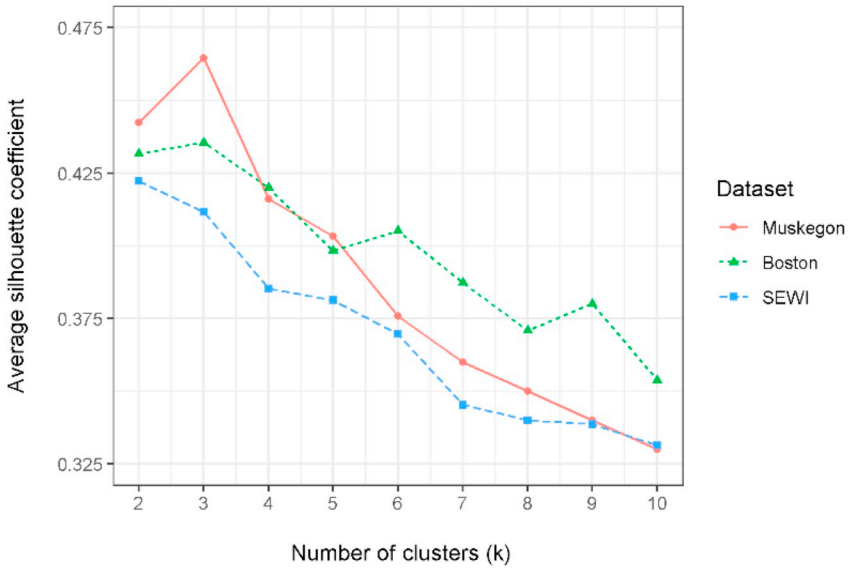

Fig. 4. Average silhouette coefficients across the number of clusters (k) for each region.

4. Results

The silhouette coefficients for each k-means calculation per study area are shown in Fig. 4. Based on the results, the goodness-of-fit of the LTM-cluster ($k = 3$) is superior to that obtained in Tayyebi et al. (2014) and the traditionally parameterized LTM ($k = 1$, Table 4). We found that the difference between applying clustering prior to the learning scheme versus not applying is significant based on the Wilcoxon signed-rank test of paired calibration statistics ($p < 0.01$). For the Muskegon case study, we also note there is an increase in performance in the PCM values from 69% for the LTM to 82% for LTM-cluster (Table 4).

Maps of the best value of k based on the largest silhouette coefficient average are shown in Fig. 5. Note that each region produces a different spatial pattern of k as it ranges from 1 to 3 (designated as Clusters 1, 2, and 3). In Muskegon, the entire eastern portion of this county had the best value of $k = 1$; the western portion had a $k = 2$ (interestingly following the US-31 highway corridor north-south and M-36 highway east-west) and the more rural areas had a value of $k = 3$ (gray). For Boston, the spatial pattern of k across the range of 1 through 3, selected again based on the largest silhouette coefficient averages, is less distinct than Muskegon; k values of 1 for example are shown as small patches arranged at the edges of the 1990 map for urban. Cluster 3 (i.e., $k = 3$) locations are all located in the upper left corner of the region. Finally, for SEWI, the map is dominated by $k = 1$ and $k = 3$ locations, with $k = 3$ following highway corridors and proximity to large 1971 urban patches and locations of $k = 1$ being in the most rural areas in 1971.

The error maps (Figs. 6–8) show the spatial pattern of agreement and disagreement which we created by overlaying the reference change maps and the urban simulated maps. The ER rates for the LTM and LTM-cluster were 37% and 31%; 33% and 32%; and 24% and 22%, for the Muskegon, Boston, and SEWI simulations. The LTM-cluster displayed a performance improvement of 19%, 3%, and 4% compared to

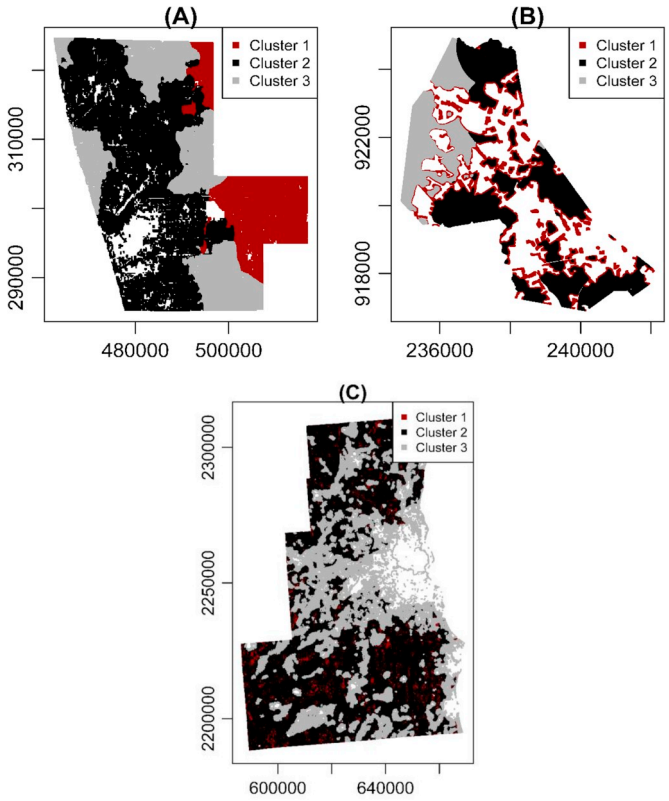

Fig. 5. Mapping of cells within clusters ($k = 3$) for Muskegon, Boston, and SEWI. Cells in white belong to the exclusionary zone (i.e., urban cells at 1978, 1990, and 1971).

Table 4
Calibration goodness-of-fit for traditionally parameterized LTM, the LTM-cluster model and the performance gain using the testing set.

Case study	LTM (%)			LTM-cluster (%)			Improvement (%)		
	ER	PCM	AUC	ER	PCM	AUC	ER	PCM	AUC
Muskegon	37.33	69.45	65.90	31.28	82.99	72.50	19.34	19.50	10.01
Boston	33.97	70.57	71.40	32.92	79.08	73.30	03.19	12.06	02.66
SEWI	24.00	79.57	82.80	22.89	82.34	83.80	04.85	03.48	01.20

Note: Improvement = (value from LTM – value from LTM-cluster)/value from LTM.

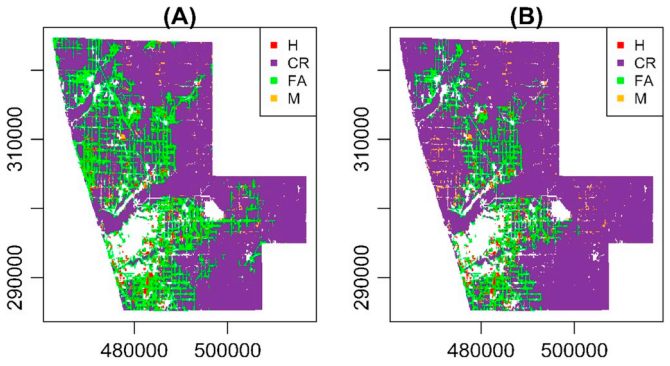

Fig. 6. Error map for (A) LTM and (B) LTM-cluster with Muskegon dataset (H = hit or true positive, CR = correct rejection or true negative, FA = false alarm or false positive, and M = miss, or false negative).

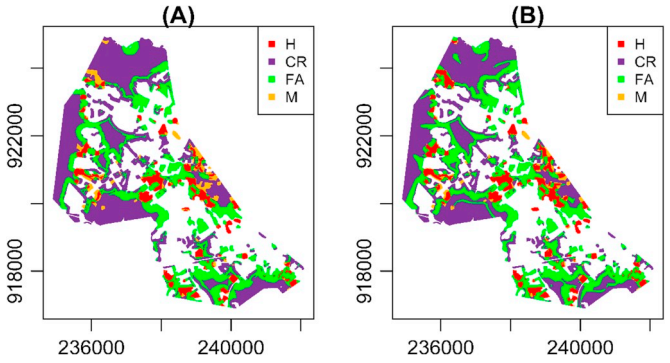

Fig. 7. Error map for (A) LTM and (B) LTM-cluster with the Boston dataset.

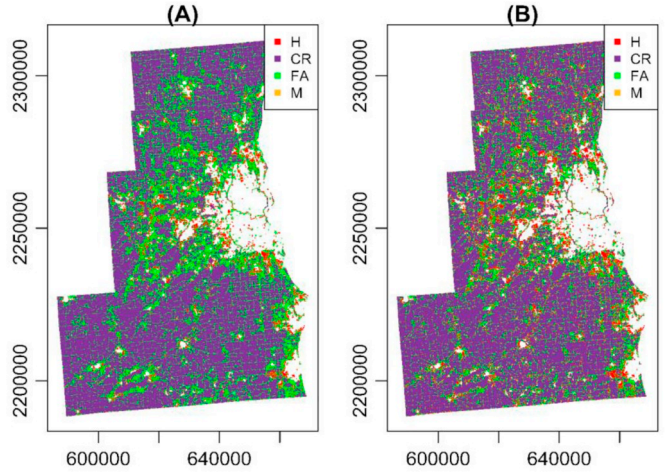

Fig. 8. Error map for (A) LTM and (B) LTM-cluster with the SEWI dataset.

the traditionally parameterized LTM model for the ER calibration statistic for the Muskegon, Boston, and SEWI simulations. AUC values also show improvements in accuracy for all three case study locations as well. Overall, the results indicate that the LTM-cluster framework provides an improvement in accuracy compared to the LTM model. Visualization of TOC curves (Figs. 9–11) also confirm that LTM-cluster outperforms the traditionally configured LTM.

Finally, we compared the computing time across the LTM and LTM-cluster model. Both models were implemented on a server (Dell PowerEdge R930, Dual Intel(R) Xeon(R) CPU E7-8891 v4 @2.80 GHz, and a Memory of 512 GB (16 × 32 Gb DDR4)). Table 5 shows that LTM-cluster model is faster than the LTM model for each dataset. Computation time decreased by 19%, 2%, and 6% for the Muskegon, Boston, and SEWI dataset.

5. Discussion

We extended the LTM using prior clustering scheme and the Spark environment. We used Spark rather than Hadoop or MapReduce, because Spark has emerged as a popular choice to implement large-scale ML applications on large datasets due to its ease in accommodating iterative learning processes (Zaharia et al., 2010, 2012). Our key findings are, first, that the clustering prior to learning scheme is an easy and effective instrument to improve the model's performance and supports the processing of large land use datasets, and, second, the improvements through LTM-cluster compared to LTM are statistically significant.

The strengths and disadvantages of the clustering prior to learning scheme are as follows:

1) Data clustering improves the performance of the model since data in each cluster are homogenous (i.e., have similar characteristics). Each cluster from the learning set is used to create a sub-model as the LTM-cluster, each having its own parameters. In addition, we demonstrated that LTM-cluster, comprising several sub-models, is superior to the basic LTM, based on a single model, in terms of several performance metrics (i.e., PCM, TOC, and AUC).
2) Clustering prior to learning scheme reduces the error rate (Table 4).
3) Dividing data across multiple clusters allows the model to scale up because data chunks from clusters are more easily maintained or processed in HPC. Each cluster is manageable in size and can be analyzed independently on separate processing units.

Although the results show that our approach is promising, there are still some limitations to our approach.

1) The Spark-based *k*-means algorithm requires one user-defined parameter. This is also the case for the majority of clustering approaches. A drawback is the difficulty in determining an optimal number of clusters (*k*). Multiple techniques exist to make that determination, and most of them are based on finding the values of *k* which balance the search of minimizing the intra-cluster distance and maximizing the inter-cluster distances. There is, however, no consensus on the “best” technique. The choice of *k* may also often rely on the researchers’ expert opinion and interpreting the goodness-of-fit of a model. In our application, silhouette indices guided our choice in an objective manner (Rousseeuw, 1987). The three datasets presented local maxima for $k = 3$ clusters, and larger values of *k* which vary with the studied dataset. We found that allowing a larger number of clusters generated many clusters with small numbers of cells. We believe that the LTM-cluster framework can be extended in several respects. Besides the Spark-based *k*-means prior learning clustering process, other clustering approaches can be explored include DBSCAN (Birant and Kut, 2007; Shen et al., 2016), ISODATA (Abbas et al., 2016; Kim and Liang, 2017), hierarchical clustering (Liu et al., 2017), and spherical *k*-means (Tunali et al., 2016).
2) We used a different set of inputs across the case studies. Despite the model improvement compared to the traditional implementation of the LTM, future studies should consider additional economic and social factors such as income levels, employment rates, and accessibility (Hagenauer and Helbich, 2018), which could greatly improve the model performance.

6. Conclusions

This study introduced a framework with an innovative model calibration sampling scheme into ML-based land change modeling. Our approach, called the “LTM-cluster” framework, is based on clustering

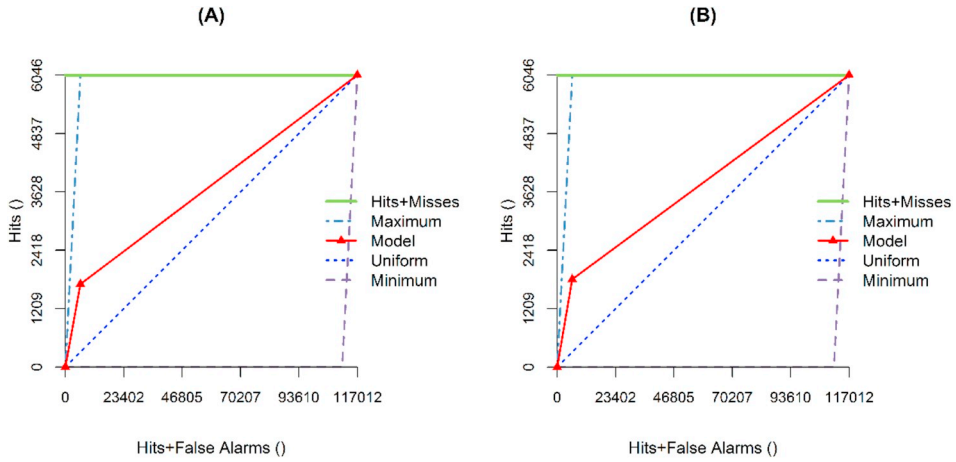

Fig. 9. TOC performance curves for the (A) LTM and (B) LTM-cluster with Muskegon dataset.

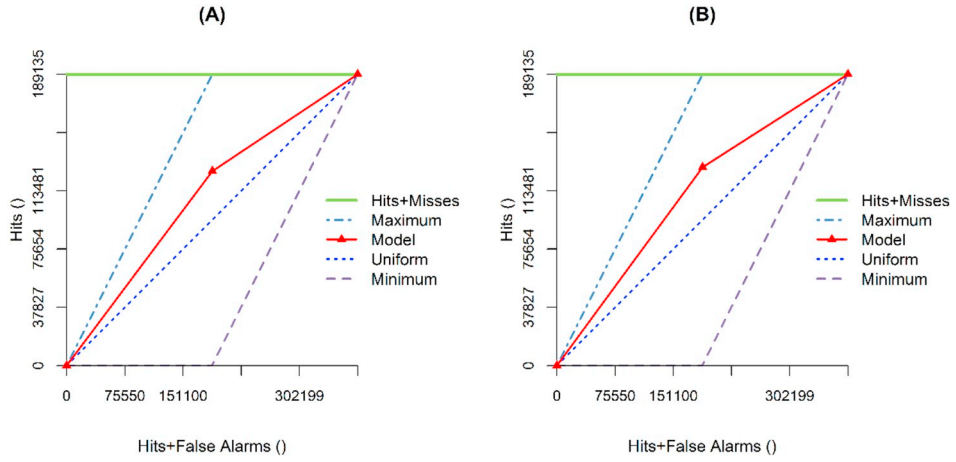

Fig. 10. TOC performance curves for the (A) LTM and (B) LTM-cluster with Boston dataset.

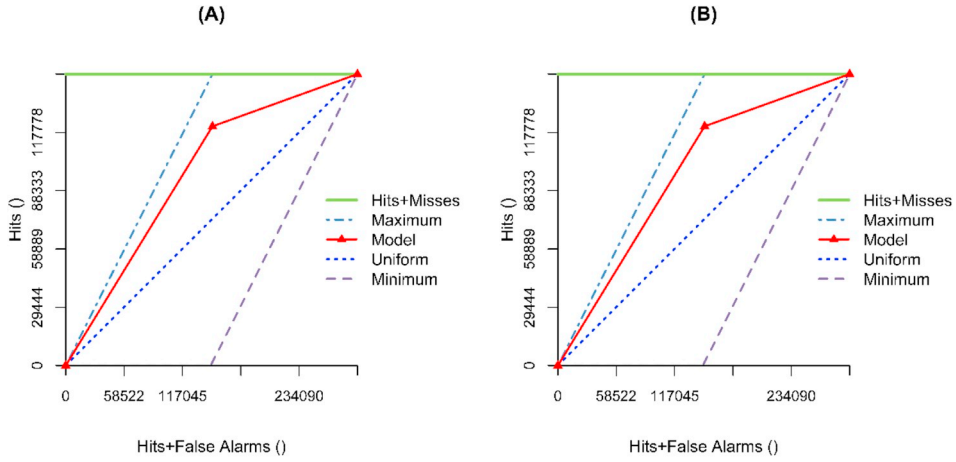

Fig. 11. TOC performance curves for the (A) LTM and (B) LTM-cluster with SEWI dataset.

using a *k*-means algorithm. Its application before the calibration step addresses the unmanageable size of the data by splitting it into clusters (i.e., groups) with similar input values. For that purpose, we implemented this framework in the Spark computing environment for large-scale data parallel processing.

We compared the performance of LTM-cluster to the traditional LTM using numerous metrics for three case studies with high variability. Results provide clear evidence that applying the clustering prior to learning scheme is significantly superior compared to the basic LTM independently of the considered performance measure. The percent correct match, the area under the curve, and the error rate improved, on average, by 9%, 11%, and 4%, respectively with LTM-cluster over all three datasets.

More research is needed to address some of the limitations of the method including the determination of the appropriate number of clusters. Testing this method in other case studies of varying size and

Table 5
Computation time (in min) for the LTM and LTM-cluster model.

Case study	LTM (in min)	LTM-cluster (in min)	Improvement (%)
Muskegon	3.57	2.86	19
Boston	3.21	3.14	2
SEWI	7.07	6.59	6

Note: Improvement = (value from LTM – value from LTM-cluster)/value from LTM.

complexity together with other ML algorithms may be useful for a more in-depth model evaluation. Findings from this study suggest that the LTM-cluster framework successfully improves the prediction accuracy and that clustering prior to learning scheme may be valuable for other applications beyond land change modeling.

Funding

This work was part of the Smart-Boundary and Smart-CA projects funded by the National Research Fund Luxembourg (FNR) and LISER research institute.

Software and data availability

The name of the software tool prototype introduced in this research is “LTM-Cluster”. The developers are the authors of the study for the conceptualization. Hichem Omrani generated the implementation of the tool in *R*. Please contact the first author for further information. Year first available: October 2018. Software required: download the *R* software from the Internet (www.r-project.org). Availability: the datasets and *R* codes of the developed model are available on the Mendeley repository (dx.doi.org/10.17632/xnxrhw4fhv.2; Omrani et al., 2018). Datasets are land use data inputs from three study areas described in this research. The datasets and the tool can be used to perform a cross-model comparison or for other purposes. Users are welcome to send their feedback to the corresponding author (hichem.omrani@liser.lu).

Acknowledgements

This research work has been partially carried out at the Center for Global Soundscapes at Purdue University. The authors thank the four reviewers as well the editor for their valuable comments.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.envsoft.2018.10.004.

Appendix

Table 2
Driving factors

	Muskegon, 1978	Boston, 1971	SEWI, 1990
1	Distance to road	Distance to transport	Distance to road
2	Distance to rivers	Distance to water	Distance to stream
3	Distance to urban	Distance to urban	Distance to wetland
4	Distance to lake	Distance to non-urban	Distance to urban
5	Distance to highways	Density of transport	Distance to forest
6	Distance to water	Density of water	Distance to park
7	–	Density of urban	Distance to agriculture
8	–	Density of non-urban	Distance to shrub
9	–	–	Density of wetland
10	–	–	Density of urban
11	–	–	Density of shrub
12	–	–	Density of forest
13	–	–	Density of agriculture
14	–	–	Elevation
15	–	–	Slope
16	–	–	Aspect

Note: Distances refer to Euclidean distances in km, slope is in %, elevation is an angle between 0° and 90, and density is a value between 0 and 1.

Table 3
Confusion matrix from the testing set (T).

Simulated map	Observed map	
	Non-urban persistence	Urban-gain
Non-urban persistence	CRs	FAs
Urban-gain	Misses	Hits
Total	Sum of non-urban persistence = CRs + Misses	Sum of urban-gain = Hits + FAs

Note: Hits or true positive, CRs: correct rejections or true negative, Misses or false positive, FAs: false alarms or false negative. PCM for urban-gain = Hits/(total of urban-gain). ER: error rate = (FA + Misses)/cardinality(T) for the LTM model. ER for the LTM-cluster = $(FA_k + Misses_k)$/sum(cardinality(T_k)) where k is the index of cluster, FA_k and $Misses_k$ are computed from the confusion matrix based on T_k subset and sum(cardinality(T_k)) = cardinality(T).

References

Abbas, A., Minallh, N., Ahmed, N., Abid, S., Khan, M., 2016. K-means and ISODATA clustering algorithms for landcover classification using remote sensing. Sindh Univ. Res. J. SURJ (Sci. Ser.) 48 (2).

Ayyadurai, P., Jayanthi, S., 2012. Ranking for web databases using svm and *k*-means algorithm. IOSR J. Comput. Eng. 8 (2), 13–18.

Azari, M., Tayyebi, A., Helbich, M., Reveshty, M.A., 2016. Integrating cellular automata, artificial neural network, and fuzzy set theory to simulate threatened orchards: application to Maragheh, Iran. GIScience Remote Sens. 53 (2), 183–205.

Basse, R.M., et al., 2014. Land use changes modelling using advanced methods: cellular automata and artificial neural networks. The spatial and explicit representation of land cover dynamics at the cross-border region scale. Appl. Geogr. 53, 160–171.

Basse, R.M., Charif, O., Bódis, K., 2016. Spatial and temporal dimensions of land use change in cross border region of Luxembourg. Development of a hybrid approach integrating GIS, cellular automata and decision learning tree models. Appl. Geogr. 67, 94–108.

Bello-Orgaz, G., Jung, J.J., Camacho, D., 2016. Social big data: recent achievements and new challenges. Inf. Fusion 28, 45–59.

Berkhin, P., 2006. A survey of clustering data mining techniques. Group. Multidimen. Data 25, 71.

Birant, D., Kut, A., 2007. ST-DBSCAN: an algorithm for clustering spatial–temporal data. Data Knowl. Eng. 60 (1), 208–221.

Blanchard, S.D., Pontius Jr., R.G., Urban, K.M., 2015. Implications of using 2 m versus 30 m spatial resolution data for suburban residential land change modeling. J. Environ. Inf. 25 (1).

Bradley, A.V., Rosa, I.M., Pontius, R.G., Ahmed, S.E., Araújo, M.B., Brown, D.G., ... Smith, M.J., 2016. SimiVal, a multi-criteria map comparison tool for land-change model projections. Environ. Model. Software 82, 229–240.

Charif, O., Omrani, H., Abdallah, F., Pijanowski, B., 2017. A multi-label cellular automata model for land change simulation. Trans. GIS 1–23.

Chiang, M.M.T., Mirkin, B., 2010. Intelligent choice of the number of clusters in *k-means* clustering: an experimental study with different cluster spreads. J. Classif. 27 (1), 3–40.

Denning, Peter J., Lewis, Ted G., January 2017. Exponential laws of computing growth. Commun. ACM 60 (1), 54–65. https://doi.org/10.1145/2976758.

Dodd, L.E., Pepe, M.S., 2003. Partial AUC estimation and regression. Biometrics 59 (3), 614–623.

Feng, Y., Liu, Y., Batty, M., 2016. Modeling urban growth with GIS based cellular automata and least squares SVM rules: a case study in Qingpu–Songjiang area of Shanghai, China. Stoch. Environ. Res. Risk Assess. 30 (5), 1387–1400.

Gopalani, S., Arora, R., 2015. Comparing Apache spark and map reduce with performance analysis using k-means. Int. J. Comput. Appl. 113 (1).

Hagenauer, J., Helbich, M., 2018. Local modelling of land consumption in Germany with RegioClust. Int. J. Appl. Earth Obs. Geoinf. 65, 46–56.

Han, J., Pei, J., Kamber, M., 2011. Data Mining: Concepts and Techniques. Elsevier.

Hashem, I.A.T., Yaqoob, I., Anuar, N.B., Mokhtar, S., Gani, A., Khan, S.U., 2015. The rise of "big data" on cloud computing: review and open research issues. Inf. Syst. 47, 98–115.

Jain, A.K., 2010. Data clustering: 50 years beyond *K*-means. Pattern Recogn. Lett. 31 (8), 651–666.

Jain, A.K., Murty, M.N., Flynn, P.J., 1999. Data clustering: a review. ACM Comput. Surv. 31 (3), 264–323.

Kanungo, T., Mount, D.M., Netanyahu, N.S., Piatko, C.D., Silverman, R., Wu, A.Y., 2002. An efficient k-means clustering algorithm: analysis and implementation. IEEE Trans. Pattern Anal. Mach. Intell. (7), 881–892.

Kim, W., Liang, S., 2017. Unsupervised classification. Int. Encycl. Geogr. People, Earth, Environ. Technol. 7. https://doi.org/10.1002/9781118786352.wbieg0271.

Liu, Y., Feng, Y., 2016. Simulating the impact of economic and environmental strategies on future urban growth scenarios in ningbo, China. Sustainability 8 (10), 1045.

Liu, Y., Phinn, S.R., 2003. Modelling urban development with cellular automata incorporating fuzzy-set approaches. Comput. Environ. Urban Syst. 27 (6), 637–658.

Liu, A.A., Su, Y.T., Nie, W.Z., Kankanhalli, M., 2017. Hierarchical clustering multi-task learning for joint human action grouping and recognition. IEEE Trans. Pattern Anal. Mach. Intell. 39 (1), 102–114.

Meng, X., Bradley, J., Yavuz, B., Sparks, E., Venkataraman, S., Liu, D., Xin, D., 2016. Mllib: machine learning in Apache spark. J. Mach. Learn. Res. 17 (34), 1–7.

Moore, N., Alagarswamy, G., Pijanowski, B., Thornton, P., Lofgren, B., Olson, J., ... Qi, J., 2012. East African food security as influenced by future climate change and land use change at local to regional scales. Climatic Change 110 (3), 823–844.

Murray, P.W., Agard, B., Barajas, M.A., 2017. Market segmentation through data mining: a method to extract behaviors from a noisy data set. Comput. Ind. Eng. 109, 233–252. https://doi.org/10.1016/j.cie.2017.04.017.

Mustafa, A., Heppenstall, A., Omrani, H., Saadi, I., Cools, M., Teller, J., 2018. Modelling Built-up Expansion and Densification with Multinomial Logistic Regression, Cellular Automata and Genetic Algorithm. Computers, Environment and Urban Systems.

Olson, J.M., Alagarswamy, G., Andresen, J.A., Campbell, D.J., Davis, A.Y., Ge, J., ... Pijanowski, B.C., 2008. Integrating diverse methods to understand climate–land interactions in East Africa. Geoforum 39 (2), 898–911.

Omrani, H., 2015. Predicting travel mode of individuals by machine learning. Trans. Res. Proc. 10, 840–849.

Omrani, H., Charif, O., Gerber, P., Awasthi, A., Trigano, P., 2013. Prediction of individual travel mode with evidential neural network model. Transport. Res. Rec.: J. Trans. Res. Board 2399, 1–8.

Omrani, H., Abdallah, F., Charif, O., Longford, N.T., 2015. Multi-label class assignment in land-use modelling. Int. J. Geogr. Inf. Sci. 29 (6), 1023–1041.

Omrani, H., Tayyebi, A., Pijanowski, B., 2017a. Integrating the multi-label land-use concept and cellular automata with the artificial neural network-based Land Transformation Model: an integrated ML-CA-LTM modeling framework. GIScience Remote Sens. 54 (3), 283–304.

Omrani, H., Abdallah, F., Tayyebi, A., Pijanowski, B., 2017b. Modelling land-use change with dependence among labels. J. Environ. Inf. 30 (2), 107–118.

Omrani, H., Helbich, M., Parmentier, B., Pijanowski, B., 2018. Land use datasets from the USA, Mendeley Data, v2. https://doi.org/10.17632/xnxrhw4fhv.2.

Pijanowski, B.C., Robinson, K.D., 2011. Rates and patterns of land use change in the Upper Great Lakes States, USA: a framework for spatial temporal analysis. Landsc. Urban Plann. 102 (2), 102–116.

Pijanowski, B.C., Pithadia, S., Shellito, B.A., Alexandridis, K., 2005. Calibrating a neural network-based urban change model for two metropolitan areas of the Upper Midwest of the United States. Int. J. Geogr. Inf. Sci. 19 (2), 197–215.

Pijanowski, B.C., Alexandridis, K.T., Mueller, D., 2006. Modelling urbanization patterns in two diverse regions of the world. J. Land Use Sci. 1 (2–4), 83–108.

Pijanowski, B., Moore, N., Mauree, D., Niyogi, D., 2011. Evaluating error propagation in coupled land–atmosphere models. Earth Interact. 15 (28), 1–25.

Pijanowski, B.C., et al., 2014. A big data urban growth simulation at a national scale: configuring the GIS and neural network based Land Transformation Model to run in a High Performance Computing (HPC) environment. Environ. Model. Software 51, 250–268.

Pontius Jr., R.G., Batchu, K., 2003. Using the relative operating characteristic to quantify certainty in prediction of location of land cover change in India. Trans. GIS 7 (4), 467–484.

Pontius Jr., R.G., Si, K., 2014. The total operating characteristic to measure diagnostic ability for multiple thresholds. Int. J. Geogr. Inf. Sci. 28 (3), 570–583.

Pontius, R.G., Thontteh, O., Chen, H., 2008. Components of information for multiple resolution comparison between maps that share a real variable. Environ. Ecol. Stat. 15 (2), 111–142.

Ray, D.K., Pijanowski, B.C., 2010. A backcast land use change model to generate past land use maps: application and validation at the Muskegon River watershed of Michigan, USA. J. Land Use Sci. 5 (1), 1–29.

Ray, D.K., Duckles, J.M., Pijanowski, B.C., 2010. The impact of future land use scenarios on runoff volumes in the Muskegon River Watershed. Environ. Manag. 46 (3), 351–366.

Rousseeuw, P., 1987. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. J. Comput. Appl. Math. 20, 53–65.

Schmidhuber, J., 2015. Deep learning in neural networks: an overview. Neural Network. 61, 85–117.

Shafizadeh-Moghadam, H., Hagenauer, J., Farajzadeh, M., Helbich, M., 2015. Performance analysis of radial basis function networks and multi-layer perceptron networks in modeling urban change: a case study. Int. J. Geogr. Inf. Sci. 29 (4), 606–623.

Shafizadeh-Moghadam, H., Tayyebi, A., Helbich, M., 2017a. Transition index maps for urban growth simulation: application of artificial neural networks, weight of evidence and fuzzy multi-criteria evaluation. Environ. Monit. Assess. 189–300.

Shafizadeh-Moghadam, H., Tayyebi, A., Ahmadlou, M., Delavar, M.R., Hasanlou, M., 2017b. Integration of genetic algorithm and multiple kernel support vector regression for modeling urban growth. Comput. Environ. Urban Syst. 65, 28–40.

Shafizadeh-Moghadam, H., Asghari, A., Tayyebi, A., Taleai, M., 2017c. Coupling machine learning, tree-based and statistical models with cellular automata to simulate urban growth. Comput. Environ. Urban Syst. 64, 297–308.

Shen, J., Hao, X., Liang, Z., Liu, Y., Wang, W., Shao, L., 2016. Real-time superpixel segmentation by DBSCAN clustering algorithm. IEEE Trans. Image Process. 25 (12), 5933–5942.

Singh, D., Reddy, C.K., 2015. A survey on platforms for big data analytics. J. Big Data 2 (1), 8.

Tang, Z., Engel, B.A., Pijanowski, B.C., Lim, K.J., 2005. Forecasting land use change and its environmental impact at a watershed scale. J. Environ. Manag. 76 (1), 35–45.

Tayyebi, A., Pijanowski, B.C., 2014a. Modeling multiple land use changes using ANN, CART and MARS: comparing tradeoffs in goodness of fit and explanatory power of data mining tools. Int. J. Appl. Earth Obs. Geoinf. 28, 102–116.

Tayyebi, A., Pekin, B.K., Pijanowski, B.C., Plourde, J.D., Doucette, J.S., Braun, D., 2013. Hierarchical modeling of urban growth across the conterminous USA: developing meso-scale quantity drivers for the Land Transformation Model. J. Land Use Sci. 8 (4), 422–442.

Tayyebi, A., Pijanowski, B.C., Linderman, M., Gratton, C., 2014. Comparing three global parametric and local non-parametric models to simulate land use change in diverse areas of the world. Environ. Model. Software 59, 202–221.

Tayyebi, A., Tayyebi, A.H., Arsanjani, J.J., Moghadam, H.S., Omrani, H., 2016. FSAUA: a framework for sensitivity analysis and uncertainty assessment in historical and forecasted land use maps. Environ. Model. Software 84, 70–84.

Tunali, V., Bilgin, T., Camurcu, A., 2016. An improved clustering algorithm for text mining: multi-cluster spherical k-means. Int. Arab J. Inf. Technol. 13 (1), 12–19.

Tuzel, O., Porikli, F., Meer, P., 2007. Human detection via classification on riemannian manifolds. In: Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on. IEEE, pp. 1–8.

Wiley, M.J., Hyndman, D.W., Pijanowski, B.C., Kendall, A.D., Riseng, C., Rutherford, E.S., ... Steen, P.J., 2010. A multi-modeling approach to evaluating climate and land use change impacts in a Great Lakes River Basin. Hydrobiologia 657 (1), 243–262.

Xin, R.S., Rosen, J., Zaharia, M., Franklin, M.J., Shenker, S., Stoica, I., 2013, June. Shark: SQL and rich analytics at scale. In: Proceedings of the 2013 ACM SIGMOD

International Conference on Management of Data. ACM, pp. 13–24.
Yang, Q., Li, X., Shi, X., 2008. Cellular automata for simulating land use changes based on support vector machines. Comput. Geosci. 34 (6), 592–602.
Zaharia, M., Chowdhury, M., Franklin, M.J., Shenker, S., Stoica, I., 2010. Spark: cluster computing with working sets. HotCloud 10 (10–10), 95.
Zaharia, M., Chowdhury, M., Das, T., Dave, A., Ma, J., McCauley, M., ... Stoica, I., 2012, April. Resilient distributed datasets: a fault-tolerant abstraction for in-memory cluster computing. In: Proceedings of the 9th USENIX Conference on Networked Systems Design and Implementation. USENIX Association 2-2.
Zhang, G., Patuwo, B.E., Hu, M.Y., 1998. Forecasting with artificial neural networks: the state of the art. Int. J. Forecast. 14 (1), 35–62.