Library Progress International
Vol.44 No.3, Jul-Dec 2024: P.3036-3054

Print version ISSN 0970 1052
Online version ISSN 2320 317X

Original Article

Available online at www.bpasjournals.com

Applying the Artificial Intelligence in forecasting the Urban growth for Al-Kut City, Iraq

Hamid D. Obead[1], Prof.Dr.Mustafa Abd Al-Jaleel[2]

[1]The Urban and Regional Planning Center for graduate studies, Baghdad University.
[2]The Urban and Regional Planning Center for graduate studies, Baghdad University.
newhamid801@gmail.com

How to cite this article: Hamid D. Obead1, Prof.Dr.Mustafa Abd Al-Jaleel (2024) Applying the Artificial Intelligence in forecasting the Urban growth for Al-Kut City, Iraq. *Library Progress International*, 44(3), 3036-3054.

Abstract
The integration of artificial intelligence (AI) with geographical and spatial data analysis enhances our understanding of the environment and addresses spatial challenges. Geospatial Artificial Intelligence (GeoAI) leverages advanced modeling and systematic visualization to monitor and predict spatial realities. This research explores the use of GeoAI in determining the directions of spatial expansion within a study area, specifically by integrating Artificial Neural Networks (ANNs) and Cellular Automata (CA) using MATLAB for data processing and analysis. The objective is to explore how these models interact to provide accurate predictions of urban growth.
Intelligent algorithms and models assist in urban planning by identifying the type, direction, and extent of future expansions. Machine learning, a branch of AI, is employed for high-accuracy classification and land use variability analysis. MATLAB software is utilized to train data and predict future land use changes. An analysis of Landsat satellite images from 2003, 2013, and 2023 was conducted using machine learning to classify land use, calculate area percentages, and illustrate the urban expansion in the study area. The findings indicate an increase in urban land from 8% in 2003 to 20% in 2023, demonstrating significant growth.

Keywords:
Artificial intelligence, data, spatial, Artifical Nueral Network, urban, Model

Introduction

Urbanization is a dynamic and complex process influenced by various factors, including population growth, economic development, and environmental conditions. As cities expand, understanding and predicting these changes becomes crucial for sustainable urban planning. Recent advancements in artificial intelligence (AI) have provided powerful tools to enhance our ability to forecast urban growth and manage spatial development effectively.[1]
Artificial Intelligence, particularly through machine learning and neural networks, has emerged as a transformative force in analyzing and interpreting spatial data. By integrating AI with geospatial data, researchers can gain deeper insights into urban expansion patterns and make informed decisions based on accurate predictions. This research explores the application of AI in forecasting urban growth for Al-Kut City, Iraq, leveraging the potential of Artificial Neural Networks (ANNs) and Constrained Cellular Automata (CA) models.[2]
A classical CA model provides a framework for understanding how urban landscapes evolve over time. It consists of a grid of cells, each representing a specific state and updating based on predefined rules and the states of neighboring cells. While traditional CA models offer valuable insights, their capabilities can be significantly enhanced by incorporating AI techniques. Machine learning algorithms excel in classifying, interpreting, and analyzing spatial data, offering high accuracy and reliability.
In this study, we utilize Landsat satellite images from 2003, 2013, and 2023 to capture temporal changes in land use and urban growth. These images serve as a foundational dataset for our analysis, which is further enriched by spatial and population data. By processing and analyzing these data using Geographic Information Systems (GIS) and advanced tools

like MATLAB, we aim to build predictive models that forecast future urban expansion.
The integration of ANNs with CA models allows us to refine predictions and improve the accuracy of urban growth forecasts. ANNs can process complex patterns in spatial data, while CA models provide a structured approach to understanding how these patterns evolve over time. This combination offers a robust methodology for predicting urban growth and supporting strategic planning.
Our research aims to contribute valuable insights into the dynamics of urban growth in Al-Kut City, Iraq. By employing AI techniques to analyze historical data and predict future trends, we seek to enhance our understanding of the factors influencing urbanization. The outcomes of this study will provide actionable recommendations for sustainable urban planning, ensuring that future expansion is managed effectively and aligns with the city's long-term goals.
In summary, this research demonstrates the potential of AI in transforming urban growth forecasting. By integrating advanced modeling techniques with spatial data analysis, we aim to offer a comprehensive approach to predicting and managing urban expansion, ultimately supporting informed decision-making and sustainable development.

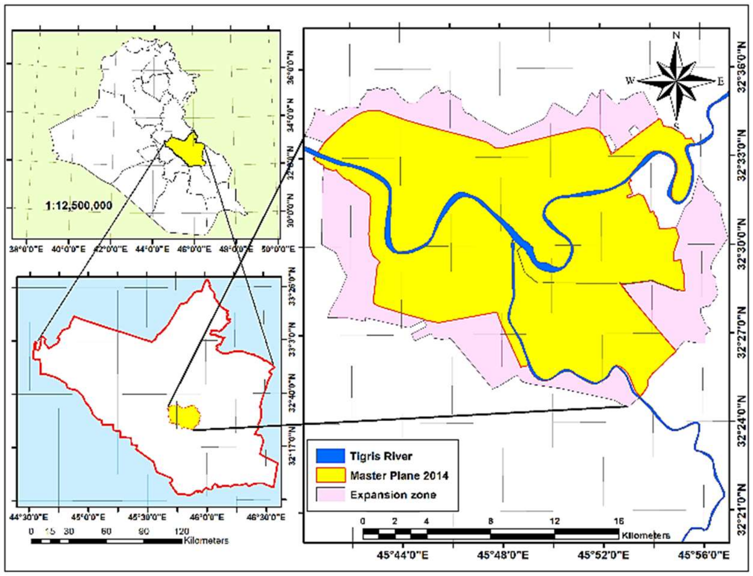
2. METERIAL AND DATASET

In this scientific research, a variety of materials and data were relied on, including satellite images of the years (2003, 2013, 2023), spatial data, population data and digital maps. Satellite images have been used as basic data for monitoring and analyzing changes in land uses and urban expansion, as these images show temporal changes in urban and non-urban lands.Spatial data represents the basis of this research, as it is processed and analyzed using Geographic Information Systems (GIS). This data is converted into digital formats (.csv) to be used in statistical modeling and analysis, facilitating the extraction of spatial patterns and the inference of relationships between various factors affecting urbanization in addition, Constrained Cellular Automata - CA models have been used that rely on a set of rules and constraints to predict changes in the urban fabric over time. These models help in analyzing factors affecting urbanization and provide accurate predictions about future trends of expansion.Advanced software tools such as MATLAB have been used to implement and analyze computational models. Loading data into MATLAB from Excel files (.csv), which contains all the criteria affecting urban growth, digital maps, population data. This data is used to train machine learning models and neural networks, enabling researchers to accurately analyze the data and make reliable predictions about urban growth.The performance of the models is evaluated by metrics such as mean square error (MSE) and correlation coefficients, to ensure the accuracy and reliability of the predictions.Ultimately, this research aims to improve our understanding of urbanization and the factors influencing it through the integration of artificial intelligence with spatial data, contributing to informed decisions in sustainable urban planning.The research will provide valuable insights into the dynamics of urban growth and support informed urban planning decisions.

4. LOCATION OF STUDY AREA:

The study area is located on the banks of the Tigris River, which divides it into two parts, with the city center forming a peninsula surrounded by the river on three sides. Notable landmarks include the famous Kut Dam and houses government offices, the Kut Textile Factory, a former military airbase now serving as an airport, and most of the colleges of Wasit University. Kut is known for its agriculture [3]. Its geographical location is of great significance due to its regional connections and climate, which affects population distribution and economic activities [4].
The study area is situated between longitudes
(E 45° 40' 20") and (E 45° 57' 00") and

Map(1) : Location of the study area



3. PREVIOUS STUDIES AND THEIR RELATIONSHIP TO CURRENT RESEARCH

Recent advancements in urban forecasting involve the use of Artificial Neural Networks (ANNs), and Cellular Automata (CA) models to enhance prediction accuracy. MATLAB offers efficient data processing, while ANNs improve predictions through their ability to recognize complex patterns.table(1)

Table 1. review of previous studies and their relevance to the current study

Previous studies	results	Relation to current research
Study of : (Najat Qader Omar,2014) (Modelling Land-use and Land-cover Changes Using Markov-CA, and Multiple Decision Making in Kirkuk City) Published in the International Journal of Scientific Research in Environmental Sciences Volume (2), Issue (1)	Have been relying on: -Markov model (CA Markov) -Multi-criteria evaluation (MCE) -Multiple regression (multi-regression) To predict urbanization trends	The Matlap program for data processing and the development of the artificial neural network(ANN) was adopted in conjunction with the cellular automation model(CA) to predict urban growth. The program is used to process data with high efficiency and provide an advanced software environment, and the artificial neural network (ANN) offers higher accuracy in predicting future changes due to its ability to learn from big data and detect complex patterns
Stydy of : (Nermin A. Shoukry , 2017) (Bulletin of the Egyptian Geographical Society) (Artificial Neural Networks Based Change Detection for Monitoring Palm Trees Plantation in Al Madinah-Saudi Arabia)	-The classification method used in this research is advanced under the supervision of artificial neural networks based on the change detection approach -The procedures for the classification of the network included the definition of criteria for the use of multi-layered feed-forward technique which is	The study of the researcher (Nermin A. Shoukry) provided technical details on how ANN can be used in image classification and analysis of changes, while the current study focused on the applications of ANN and CA in urban forecasting and analysis, focusing on the accuracy and flexibility of these models compared to traditional methods. Both studies showed how modern technologies in artificial intelligence can improve our understanding and analysis of complex data in different areas

	based on the standard backpropagation algorithm.	
Study of : (Amani A. Khalil) (Analytical study on the growth of Najaf city by adopting spatial modeling tools) Master Thesis submitted to the urban and regional planning Center for graduate studies at Baghdad University)	-One of the results of the study is that the Idrisi Selva program does not take into account social ,economic, historical and tourist factors of the city, but depends on the actual uses of the land and its area - the modern ability to easily access the Landsat archive data via the internet increases the use of these images in a wide range of fields.	The combined socio-economic model (Tietenberg Model) and(ANN) shows great potential for predicting the dynamics of urbanization. Performance evaluation criteria, such as high correlation coefficients ,minimum verification errors and optimal gradient values ,support the efficiency and reliability of the model The researcher's current study supports this trend, as he showed through his use of Google Earth Engine platform in image classification that the ability to easily access Landsat archive data significantly contributes to improving the accuracy of classification and expanding the use of images in various scientific applications."

5. GEOSPATIAL ARTIFICIAL INTELLIGENCE (GEOAI)

It uses machine learning techniques to go beyond traditional statistical methods by deciphering location-based information and offering more effective and accurate solutions to specific geographic issues. GeoAI consists of independent software algorithms that can be integrated into geographic information systems (GIS) or remote sensing systems, and uses a dataset to reach conclusions without external influence. Geographic modeling combines artificial intelligence and geospatial data to extract insights, forecasts and automate spatial analysis, enabling accurate analysis of complex spatial relationships [5]

GeoAI modeling has several main characteristics:

5.a: Machine learning integration: it makes use of algorithms such as neural networks, strong support methods and decision trees to analyze spatial data and derive patterns and relationships [6]

5.b: Spatial data processing: techniques such as convolutional and iterative neural networks are used to process image and satellite data and analyze time series [7]

5.c: Predictive analytics: provides predictive models to predict spatial phenomena such as land cover changes, urban growth and environmental pollution [8]

5.d:Interpretability and clarification: interpretable AI techniques are integrated to increase the transparency of the model and enable users to understand the factors influencing the predictions

5.e:High efficiency: efficiently manages large geographical areas and heterogeneous data through parallel processing, distributed calculations and cloud-based infrastructure [9]

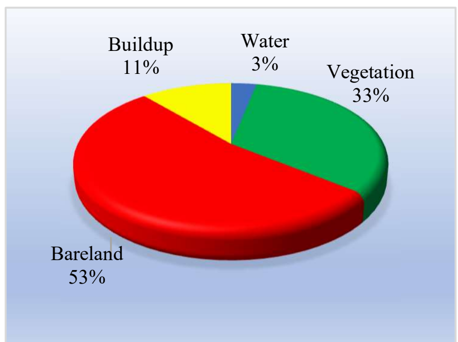
6. The Programmatic Classification Technique

Using JavaScript, which is relevant to the Google Earth Engine (GEE) platform, a land cover classification for the study area was created by generating a training sample. High-resolution base map images from Google Maps were used to identify the four land cover classes. With these training samples, and using drawing tools in the Code Editor, we train a classifier (smileRandomForest) to build a model. This model is then applied to all pixels in the image.The urban landscape of the study area was classified into four categories (urban, Vegetation, barren land, and water) and analyzed for the years 2003, 2013, and 2023. In 2003,Figure.(1.a) barren land were the most dominant land use, accounting for approximately 53%. This was followed by green and agricultural areas at 33%, which were concentrated in the northeast and partially in the southeast. Built-up areas made up 11%, primarily concentrated around the city center due to the non-arable nature of the barren land and their soil salinity.By 2023, Figure.(1.b) barren land had increased to 63% of the land use, followed by agricultural areas at 24%. Built-up areas had expanded significantly, especially in the north and northeast of the study area, with an increase of over 5,000 hectares, representing about 20%. Agricultural lands had decreased in favor of barren land, which increased to 47% compared to 31%. Water resources maintained a near-constant percentage of 4%, but declined to 2% in the most recent year. Figure.(1.c)& table(2)

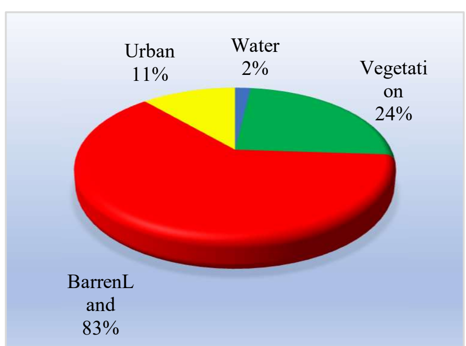
Table (2): Land Use Areas and Percentages of the Study Area for the Years 2003, 2013, and 2023

landuse	2003 Hectar	%	2013 Hectar	%	2023 Hectar	%
Urban Area	3232.8	8	4300.0	11	7464.1	20
Vegetation	7764.0	21	9114.7	24	11575.9	31
Barrenland	25551	68	23609	63	17965	47
Water Body	1189.4	3	713.53	2	732.3	2
Total	37737	100	37737	10	37737	10

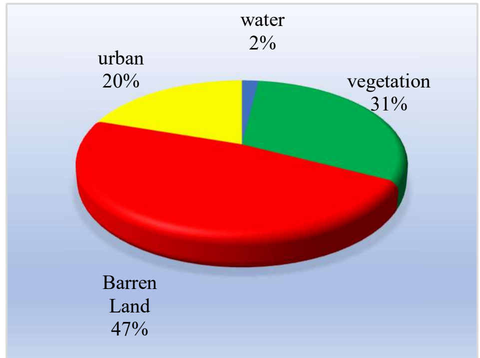
Source: Compiled by Authors



(a) Percentage landuse in 2003



(b) Percentage landuse in 2013



(c) Percentage landuse in 2023

Figure(1) : Percentage of Land Use of the Study Area for the years (2003,2013 and 2023)

Here is a workflow diagram that illustrates the steps involved in classifying the study area as "supervised classification."Figure(2)

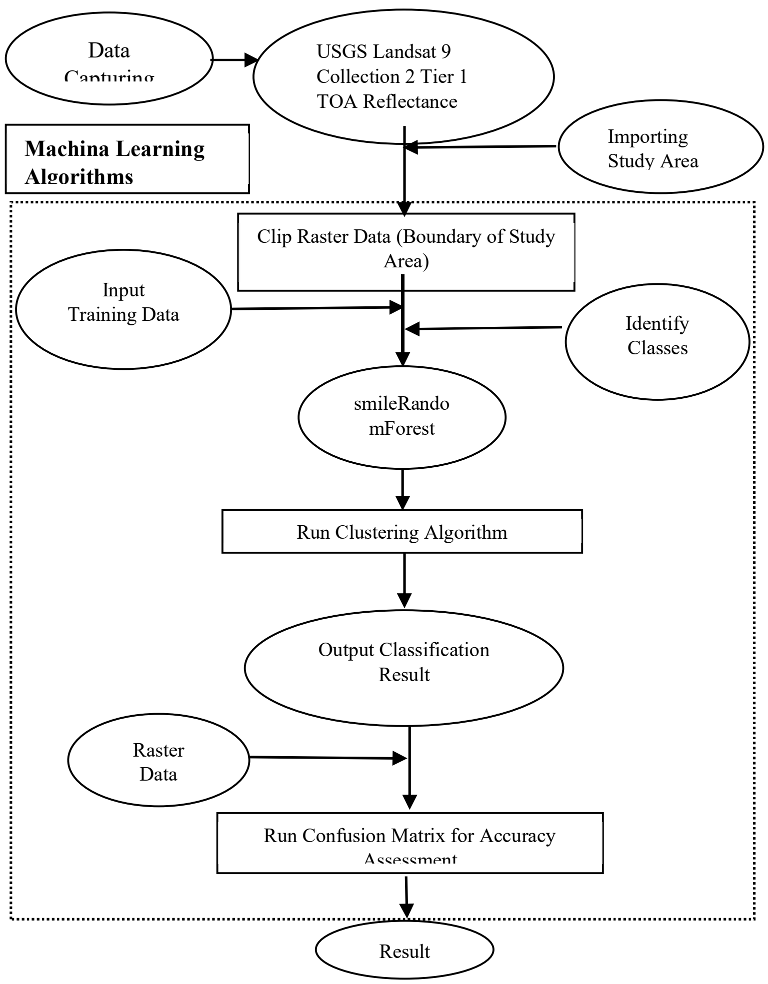


Figure (2): Flowchart of the Programmatic Classification proccess

This workflow outlines the process of supervised classification using machine learning for land cover mapping. It begins with data collection and importing the study area's boundaries. The raster data is clipped, and training data is used to identify land cover classes. A machine learning algorithm, such as Random Forest, is applied to classify the area. The results are then validated using a confusion matrix to assess accuracy.

7.Code Implementation:

8.a: Providing the Dataset to the Program :
The MATLAB program is supplied with the available data from an Excel file (.csv) that includes three sheets: The first sheet contains all the criteria affecting urban growth, the second contains the satellite images of the area for the years 2003, 2013, and 2023 in digital format, and the third sheet includes population data and growth rates (see Appendix A) The data is read from the Excel file and stored in a MATLAB table.
8.b: Assuming the Existence of a Suitability Column in the Factors Set :
The code aims to extract features from the (factors_data) table that represent the suitability of certain areas. A variable (feature_columns_factors) is defined as a set of column names, and the values from these columns are extracted and stored in a new matrix (X_factors).
8.c: Reading the Factors Affecting Urban Expansion from the Second Sheet :
The code uses the (readtable) function to read the data from the Excel file and extract specific columns that contain information on classifying areas as urban or non-urban for the years 2003, 2009, 2013, and 2023, storing them in a matrix (X_urban).
8.d: Reading Population Data from the Third Sheet
After reading, the relevant columns are extracted and prepared for use in analysis or modeling. The data is stored in the variable (X_population), and the population values are stored in the array (y_population).
8.e: Loading Spatial Suitability Criteria Maps
Maps for various variables are loaded to assess the suitability of the area for expansion using the function (imread) to convert them into arrays for programmatic processing.
8.f: Obtaining Basic Image Dimensions
The dimensions of the images are obtained using the function (size) and the images are resized to match the basic dimensions using (imresize).
8.g: Adjusting Dimensions of (X_population) and (X_urban)
The data dimensions are checked for consistency and adjusted if necessary. Then, the features of X_population are merged into a new variable (X) and the values of y_population into a variable (y).
8.h: Preparing the Artificial Neural Network Structure
An artificial neural network is prepared and trained using input data (X) and output data (y). The number of nodes in the different layers is determined, and the network is trained using the function (train).
8.i: Preparing the Years for Prediction
A matrix containing the years to be predicted is defined, and the size of this matrix is displayed.
8.j: Linear Regression Model for Training Data
A linear regression model is used to build a prediction model using the function (fitlm) to estimate urban population based on historical data.
8.k: Implementing the Constrained Cellular Automata Model
The cellular automata model is implemented to predict changes in urban population distribution based on a specified annual growth rate. The code involves setting up the initial network, calculating annual population growth, and displaying the results graphically.
8.l: Applying Cellular Automata Rules and Updating the Initial Network
Cellular automata rules are applied to the updated network over several iterations. Data related to urban density is collected, and the final average urban density is calculated using the array (totalUrbanDensity).
8.m: Extracting Relevant Population Data for Specified Years (1977-2022)
Population data is prepared for use in subsequent analyses or models by extracting specific population data for the years between 1977 and 2022 from the spreadsheet and verifying the consistency of the years.

8- STRUCTURING AND SIMULATING THE TIETENBERG MODEL

The Tietenberg Model is integrated into the ANN-UrbanCA model to project future demand for urban spaces by calculating the number of new urban cells generated during each iteration. Originally a resource economics model, Tietenberg's framework, as applied in this study, addresses sustainable land use during urbanization. It factors in population growth, economic development, and land-use policies to predict future urban area requirements, aligning with sustainable development principles. The model simulates urban area evolution over time, ensuring sustainable land management.
The logistic regression function used for future population growth is given by:[10]

$$Xt= \frac{Xm}{1+(\frac{Xm}{Xo}-1)e^{-rt}}$$

Where :
X_0: Initial population at time t=0
Xt: Population at time t
r: Population growth rate
Xm: Population carrying capacity
e: The base of the natural logarithm (approximately 2.7)

9-DETERMINING THE INITIAL PARAMETERS FOR THE TIETENBERG MODEL

The model describes how a population grows in a limited environment, taking into account the initial population, the environment's carrying capacity, and the population growth rate. The logistic growth model is given by equation above. See Table (3)

Table (3): Determination of Tietenberg Model Parameters

The code	The description
X0 = 100000; % Initial population	The initial population at time (t = 0)
Xm = 500000; % Carrying capacity	Carrying capacity of the population size
r = 0.02; % Population growth rate	r = 0.02, the population growth rate
time_periods = 50; % Number of time periods for the forecast	50 represents the time period for which we want to predict population growth.
population_forecast=logistic_population_growth (X0,Xm,r,1:time_periods);	Predictions for population size over 50 periods

When the population reaches 500,000, its growth rate will slow down and eventually stop. The number of time periods refers to the intervals over which we calculate population forecasts, indicating the duration for which we want to predict growth. The `logistic_population_growth` function is used to calculate growth based on the logistic model. The `population_forecast` array contains the predicted population sizes for each time period from 1 to 50. In essence, initial values are used to compute forecasts for each time period using the logistic equation, with results stored in the `population_forecast` array. This array shows the population projections over 50 periods based on the logistic growth model.

10-MODEL PERFORMANCE EVALUATION

Figure (3) shows the performance of the neural network model across several stages of training, using Mean Squared Error (MSE) as a performance metric over 198 epochs.

The best performance on the validation data was achieved at epoch 192, where the Mean Squared Error (MSE) was 0.035751. This indicates that the model was at its peak accuracy on the validation data during this epoch [11].

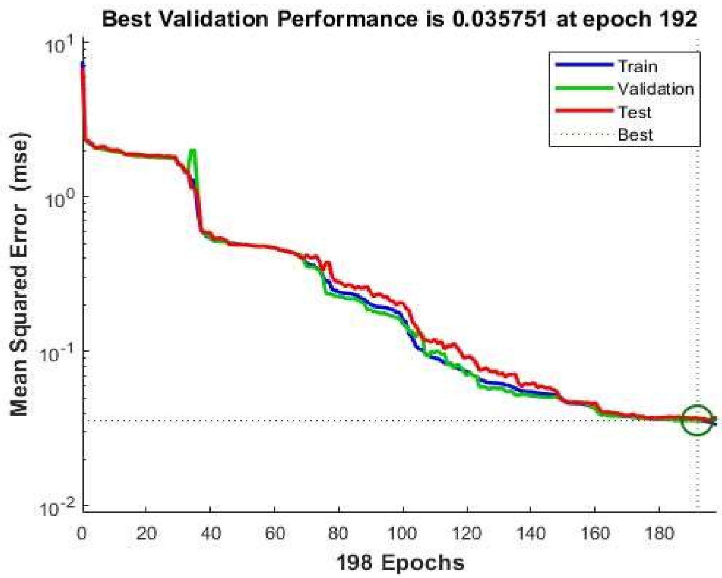


Figure (3): Error Graph of the Artificial Intelligence Neural Network Model

The green line (Validation) and the red line (Test) suggest that the model is not only able to learn the training data well (blue line), but also performs well on data that was not used in the training process.If the MSE values on the validation and test data are low and close to the MSE value on the training data, this indicates that the model is capable of generalizing well to new data. In this plot, it is shown that the three lines (blue, green, and red) converge towards the end of training, which indicates that the model is not overfitting and retains its ability to generalize

11.Correlation Coefficients

The correlation coefficient measures the strength of the relationship between two variables, in this case, the predicted and actual patterns of urban growth. The values range from -1 to +1, with higher values indicating a stronger relationship.

- Training Set (R=0.99801): The model predicted urban growth with very high accuracy on the training data.
- Validation Set (R=0.99741): The model also performed with high accuracy on unseen data.

These high correlation values show a strong linear relationship, meaning the model effectively captures the dynamics of urban growth [11].Figure.(4)

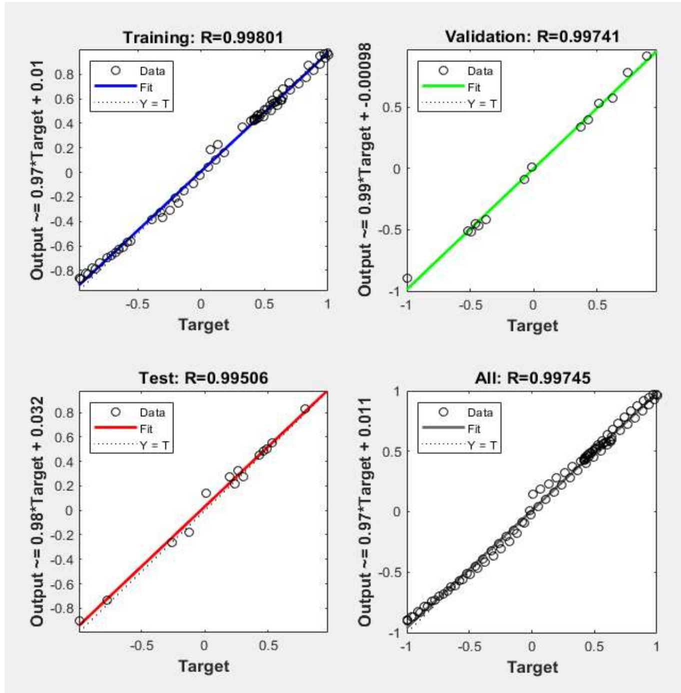


Figure (4): Correlation Coefficients

12. CONCLUSION

1. Artificial Intelligence (AI) techniques will not eliminate human skills, but rather, a blend of machine skills and human expertise will emerge.
2. The potential of Geospatial artificial intelligence and planning becomes evident in its ability to analyze comprehensive and complex datasets, automate and enhance planning processes, and provide more accurate predictions and insights into urban and environmental dynamics.
3. The adoption of AI enhances automation, which results in a reduced need for employees to handle routine and repetitive tasks. Concurrently, there is an increased demand for creative human expertise with deep insights to address more complex issues.
4. Classifying the study area using machine learning (Google Earth Engine) may contribute to understanding the distribution of various land uses in the studied area, which can be beneficial for land use planning and the development of urban and agricultural areas in the future.
5. AI methods, such as Cellular Automata, allow for the simultaneous simulation of different land cover classes, providing insights into how decision-makers perceive the factors and constraints affecting urban development and prioritize them.
6. Through performance evaluation, it can be said that the AI model demonstrates good generalization and prediction capabilities on new data, which is important for applying the model in forecasting future urban growth patterns. Low verification errors indicate the model's reliability and stability in predictions, enhancing its value as a predictive tool in the field of urban growth.
7. High correlation coefficients confirm the model's ability to accurately simulate complex urban processes, supporting its application in urban planning and policy formulation. This means that the model can be relied upon to understand how cities evolve and grow over time.

13. RECOMMENDATION

1. Sustainable urban planning requires the integration of advanced technology to enhance environmental, economic, and social quality. It relies on fifth-generation technologies and neural networks to efficiently manage city resources and improve smart infrastructure. Hybrid intelligent systems and neural systems help enhance decision-making and provide

tailored services to residents. Neural systems contribute to predicting environmental issues and offering proactive solutions, which boosts the overall sustainability of the city and ensures the long-term well-being of its inhabitants.
2. It is preferable to use the model trained until epoch (Epoch = 192), as it achieved the best performance on validation data.
3. Statistical techniques such as regression analysis or Principal Component Analysis (PCA) can be used to provide deeper insights into the behavior of the artificial intelligence model discussed in the study, thereby assisting in more effective data analysis.

14. THE REFERANCES

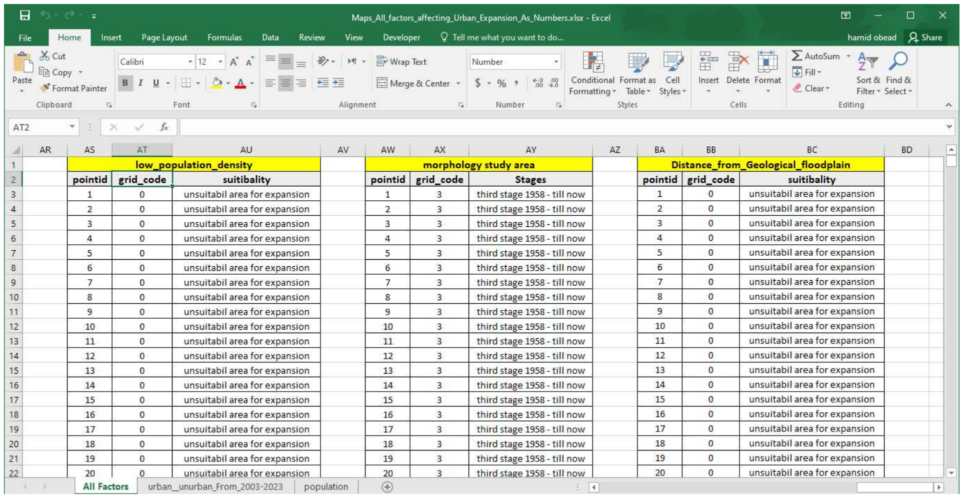
[1] Ning Wu , and Elisabete A. Silva,(2010), "*Artificial Intelligence Solutions for Urban Land Dynamics*",Journal of Planning Literature, 24(3) 246-265, DOI: 10.1177/0885412210361571
[2] Aburas, Maher Milad and Ho, Yuek Ming,(2018)," *Spatio-temporal simulation of future urban growth trends using an integrated CA-Markov model*", Arabian Journal of Geosciences.
[3] Al-Tamimi, Najla Jasim Hameed. (2009). "*The Spatial Organization of Urban Settlements in Wasit Governorate.*" Master's Thesis, University of Baghdad, Urban and Regional Planning Center for Postgraduate Studies.p3
[4] Al-Miyahi, Abdul Amir. (2000). "*A Comparative Study of the Functional Region of the Cities of Kut and Hilla*". PhD Dissertation, University of Baghdad ,College of Arts.p152
[5] Mai, Gengchen, Chris Cundy, Kristy Choi, Yingjie Hu, Ni Lao, Stefano Ermon.,(2022), "*Towards a foundation model for geospatial artificial intelligence (vision paper)*." Proceedings of the 30th International Conference on Advances in Geographic Information Systems,SIGSPATIAL,Seattle, WA, USA
[6] Liu, Pengyuan, Yan Zhang, and Filip Biljecki.,(2023), "Explainable spatially explicit geospatial artificial intelligence in urban analytics." ,Environment and Planning B: Urban Analytics and City Science ,doi:23998083231204689.
[7] Yu, Jia, Jinxuan Wu, and Mohamed Sarwat.,(2015) "Geospark: A cluster computing framework for processing large-scale spatial data." Proceedings of the 23rd SIGSPATIAL international conference on advances in geographic information systems.
[8] Shmueli, Galit, and Otto R. Koppius,(2011), "Predictive analytics in information systems research.", Management Information Systems (MIS) quarterly .
[9] Iyer,CVKrishnakumar,Swetava Ganguli, and Vipul Pandey,(2023),"Perspectives on Geospatial Artificial IntelligencePlatforms for Multimodal Spatiotemporal Datasets.", Advances in Scalable and Intelligent Geospatial Analytics
[10]Tietenberg, T.(1992), Environmental and natural resource economics. New York, New York: Harper Collins
[11]Shabib, Hader Ali Abdel Badie , (2021), "a proposed model for assessing the efficiency of strategic performance using artificial intelligence systems for medium, small and microenterprises in Egypt : an applied Study", Journal of financial and business studies, third issue

Official institutions:
-Misitry of water resources , general state on commssion / Geographic Information Systen Section

APPENDIX(A)

A.1: THE DATASET USED IN THE ARTIFICIAL INTELLIGENCE CODE

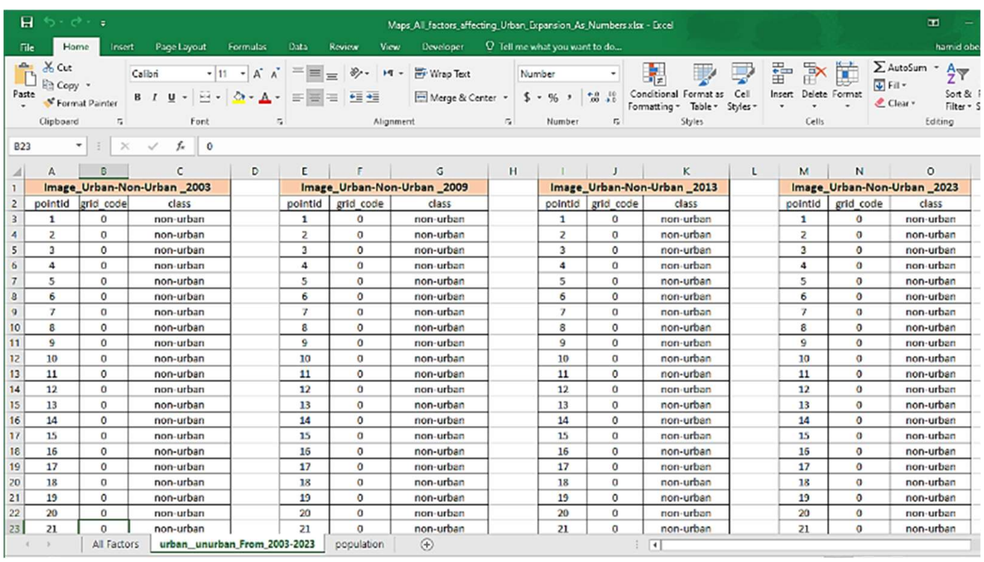
variety of spatial data was utilized, processed, and analyzed using Geographic Information Systems (GIS). The first sheet of the data is presented in the Excel file. Figure (A1) shows the dataset of factors influencing urban expansion after converting their maps into digital data. This approach is essential for transforming the initial geographic data into a form that can be analyzed statistically and modeled using advanced techniques.



low_population_density			morphology study area			Distance_from_Geological_floodplain		
pointid	grid_code	suitibality	pointid	grid_code	Stages	pointid	grid_code	suitibality
1	0	unsuitabil area for expansion	1	3	third stage 1958 - till now	1	0	unsuitabil area for expansion
2	0	unsuitabil area for expansion	2	3	third stage 1958 - till now	2	0	unsuitabil area for expansion
3	0	unsuitabil area for expansion	3	3	third stage 1958 - till now	3	0	unsuitabil area for expansion
4	0	unsuitabil area for expansion	4	3	third stage 1958 - till now	4	0	unsuitabil area for expansion
5	0	unsuitabil area for expansion	5	3	third stage 1958 - till now	5	0	unsuitabil area for expansion
6	0	unsuitabil area for expansion	6	3	third stage 1958 - till now	6	0	unsuitabil area for expansion
7	0	unsuitabil area for expansion	7	3	third stage 1958 - till now	7	0	unsuitabil area for expansion
8	0	unsuitabil area for expansion	8	3	third stage 1958 - till now	8	0	unsuitabil area for expansion
9	0	unsuitabil area for expansion	9	3	third stage 1958 - till now	9	0	unsuitabil area for expansion
10	0	unsuitabil area for expansion	10	3	third stage 1958 - till now	10	0	unsuitabil area for expansion
11	0	unsuitabil area for expansion	11	3	third stage 1958 - till now	11	0	unsuitabil area for expansion
12	0	unsuitabil area for expansion	12	3	third stage 1958 - till now	12	0	unsuitabil area for expansion
13	0	unsuitabil area for expansion	13	3	third stage 1958 - till now	13	0	unsuitabil area for expansion
14	0	unsuitabil area for expansion	14	3	third stage 1958 - till now	14	0	unsuitabil area for expansion
15	0	unsuitabil area for expansion	15	3	third stage 1958 - till now	15	0	unsuitabil area for expansion
16	0	unsuitabil area for expansion	16	3	third stage 1958 - till now	16	0	unsuitabil area for expansion
17	0	unsuitabil area for expansion	17	3	third stage 1958 - till now	17	0	unsuitabil area for expansion
18	0	unsuitabil area for expansion	18	3	third stage 1958 - till now	18	0	unsuitabil area for expansion
19	0	unsuitabil area for expansion	19	3	third stage 1958 - till now	19	0	unsuitabil area for expansion
20	0	unsuitabil area for expansion	20	3	third stage 1958 - till now	20	0	unsuitabil area for expansion

Figure (A1) shows the first sheet of the dataset containing the code data in an Excel file(csv.).

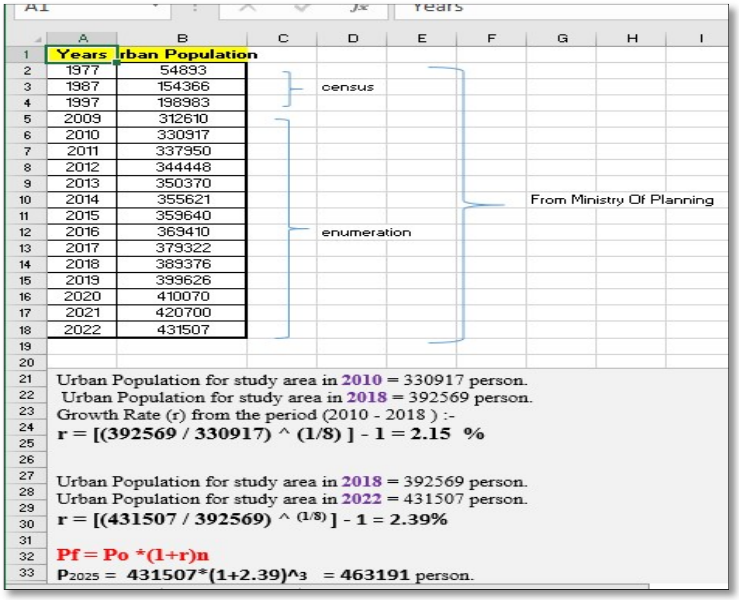
As shown in the figure below, the data conversion process involves multiple stages of verification and modification to ensure the accuracy of the input data. GIS tools were relied upon to convert all factors, allowing researchers to perform comprehensive and precise analyses that can contribute to enhancing our understanding of spatial patterns and land use. Satellite imagery is used as a primary source of spatial data that aids researchers and urban planners in monitoring and analyzing changes in land use and urban expansion. Figure (A2) shows a sample of the data resulting from the process of converting satellite images into digital data, where the satellite images were transformed into tabular data (.csv) that records changes in urban and non-urban conditions at specific points over the years (2003, 2009, 2013, and 2023). This data is used to analyze the factors influencing urban expansion and to improve the accuracy of predictions in urban models



Image_Urban-Non-Urban _2003			Image_Urban-Non-Urban _2009			Image_Urban-Non-Urban _2013			Image_Urban-Non-Urban _2023		
pointid	grid_code	class	pointid	grid_code	class	pointid	grid_code	class	pointid	grid_code	class
1	0	non-urban	1	0	non-urban	1	0	non-urban	1	0	non-urban
2	0	non-urban	2	0	non-urban	2	0	non-urban	2	0	non-urban
3	0	non-urban	3	0	non-urban	3	0	non-urban	3	0	non-urban
4	0	non-urban	4	0	non-urban	4	0	non-urban	4	0	non-urban
5	0	non-urban	5	0	non-urban	5	0	non-urban	5	0	non-urban
6	0	non-urban	6	0	non-urban	6	0	non-urban	6	0	non-urban
7	0	non-urban	7	0	non-urban	7	0	non-urban	7	0	non-urban
8	0	non-urban	8	0	non-urban	8	0	non-urban	8	0	non-urban
9	0	non-urban	9	0	non-urban	9	0	non-urban	9	0	non-urban
10	0	non-urban	10	0	non-urban	10	0	non-urban	10	0	non-urban
11	0	non-urban	11	0	non-urban	11	0	non-urban	11	0	non-urban
12	0	non-urban	12	0	non-urban	12	0	non-urban	12	0	non-urban
13	0	non-urban	13	0	non-urban	13	0	non-urban	13	0	non-urban
14	0	non-urban	14	0	non-urban	14	0	non-urban	14	0	non-urban
15	0	non-urban	15	0	non-urban	15	0	non-urban	15	0	non-urban
16	0	non-urban	16	0	non-urban	16	0	non-urban	16	0	non-urban
17	0	non-urban	17	0	non-urban	17	0	non-urban	17	0	non-urban
18	0	non-urban	18	0	non-urban	18	0	non-urban	18	0	non-urban
19	0	non-urban	19	0	non-urban	19	0	non-urban	19	0	non-urban
20	0	non-urban	20	0	non-urban	20	0	non-urban	20	0	non-urban
21	0	non-urban	21	0	non-urban	21	0	non-urban	21	0	non-urban

Figure (A2): The second sheet of the dataset for code data in the Excel file (.csv).

Constrained Cellular Automata (CA) models are effective tools for analyzing data and predicting future changes for sustainable development. In our study, we used the CA model to forecast urban population growth using demographic data from the Ministry of Planning. This data, shown in Figure (A3), includes past population counts and calculated growth rates. The model was implemented using MATLAB for its flexibility in handling complex computational models and data analysis.



Years	Urban Population
1977	54893
1987	154366
1997	198983
2009	312610
2010	330917
2011	337950
2012	344448
2013	350370
2014	355621
2015	359640
2016	369410
2017	379322
2018	389376
2019	399626
2020	410070
2021	420700
2022	431507

census

enumeration

From Ministry Of Planning

Urban Population for study area in 2010 = 330917 person.
Urban Population for study area in 2018 = 392569 person.
Growth Rate (r) from the period (2010 - 2018) :-
r = [(392569 / 330917) ^ (1/8)] - 1 = 2.15 %

Urban Population for study area in 2018 = 392569 person.
Urban Population for study area in 2022 = 431507 person.
r = [(431507 / 392569) ^ (1/8)] - 1 = 2.39%

Pf = Po *(1+r)n
P_{2025} = 431507*(1+2.39)^3 = 463191 person.

Figure (A3):Third sheet of the dataset in the Excel (.csv) file.

A.2: THE FINAL AND EXECUTABLE PREDICTIVE ARTIFICIAL INTELLIGENCE CODE IN MATLAB

```
% Load file
filename = 'Maps_All_factors_affecting_Urban_Expansion_As_Numbers.xlsx';

% Read All factor data from the first sheet
factors_data = readtable(filename, 'Sheet', 'All Factors');
disp('Successfully read All Factors data.');
% Assuming factors_data has column for suitability
feature_columns_factors = {'suitibality', 'suitibality_1', 'suitibality_2', 'suitibality_3', 'suitibality_4', 'suitibality_5',
'suitibality_6', 'suitibality_7', 'suitibality_8', 'suitibality_9', 'suitibality_10', 'suitibality_11', 'suitibality_12' };
X_factors = factors_data{:, feature_columns_factors};

% Read factors affecting urban expansion from the second sheet
urban_data = readtable(filename, 'Sheet', 'urban__unurban_From_2003-2023');
disp('Successfully read urban__unurban_From_2003-2023 data.');
% Extract relevant features from urban data
% Var4 represent class of Urban-Non-Urban 2003
% Var4 represent class of Urban-Non-Urban 2009
% Var4 represent class of Urban-Non-Urban 2013
% Var4 represent class of Urban-Non-Urban 2023
feature_columns_urban = { 'Var4', 'Var8', 'Var12', 'Var16'};
X_urban = urban_data{:, feature_columns_urban};

% Read Population data from the third sheet
population_data = readtable(filename, 'Sheet', 'population');
disp('Successfully read population data.');
% Extract relevant features from population data
% Var1 represents Years & Var2 Population
feature_columns_population = {'Var1'};
output_column_population = {'Var2'};
X_population = population_data{:, feature_columns_population};
```

```
y_population = population_data{:, output_column_population};

% Load suitability maps
soil_mixture_index = imread('soil_mixture_index.png');
low_land_value = imread('low_land_value.png');
land_surface_temperature = imread('land_surface_temperature.png');
geological_floodplain = imread('geological_floodplain.png');
distance_to_street = imread('distance_to_street.png');

% Adjust dimensions of X_population and X_urban if necessary
if size(X_population, 1) ~= size(X_urban, 1)
   % If dimensions are not consistent, adjust X_urban to match the number of rows in X_population
   X_urban = X_urban(1:size(X_population, 1), :);
end

% Combine features from both datasets
X = [X_population]; y = y_population;

% Set up the ANN architecture
inputSize = size(X, 2);
hiddenLayerSize = 10;
outputSize = 1;
net = feedforwardnet(hiddenLayerSize);
net = configure(net, X', y');
net.trainParam.showWindow = true;
net.trainParam.epochs = 100;
net = train(net, X', y');
disp('Artificial Neural Network (ANN) trained successfully.');
% save('trained_ann_model.mat', 'net');

% Define years for forecasting
years_to_predict = [2025, 2030, 2035, 2040];

% Display the size of the years array
disp('Size of years array:');
disp(size(years_to_predict));

% Fit a linear regression model to the training data
mdl = fitlm(X_population, y_population);

% Predict urban population using the fitted linear regression model
urbanPopulationPredicted = predict(mdl, X_population);

% Convert urbanPopulationPredicted to a column vector
urbanPopulationPredicted_column = urbanPopulationPredicted(:);

% Use the latest available population as the initial urban population
initialUrbanPopulation = y(end);

% Constrained Cellular Automata (CA) Model Implementation
numIterations = 10;
gridSize = 100;
initialDensity = initialUrbanPopulation / gridSize^2;
birthThreshold = 3;
```

```
survivalThreshold = 2;
initialGrid = rand(gridSize) < initialDensity;

% Initialize variables
urbanPopulationForecast = zeros(1, length(years_to_predict) + 1);
urbanPopulationForecast(1) = initialUrbanPopulation;
annualGrowthRate = 1.5; % a growth rate of 1.5% per year


% Forecast urban population for each year
for i = 1:length(years_to_predict)
% Calculate the annual increase based on the growth rate
   annualIncrease = urbanPopulationForecast(i) * (annualGrowthRate / 100);

% Update the forecasted population for the next year
   urbanPopulationForecast(i + 1) = urbanPopulationForecast(i) + annualIncrease;
end

% Display the urban population forecast
disp('Urban Population Forecast with Annual Increase:');
disp([2010, years_to_predict]);
disp([initialUrbanPopulation, urbanPopulationForecast(2:end)]);

% Plot the urban population forecast
figure;
plot([2010, years_to_predict], [initialUrbanPopulation, urbanPopulationForecast(2:end)], '-o');
title('Urban Population Forecast with Annual Increase');
xlabel('Year');
ylabel('Urban Population');
grid on;

% Plot the integrated urban growth forecast
figure;

% Apply cellular automaton rules and update initialGrid
updatedGrid = initialGrid;
for iteration = 1:numIterations
   updatedGrid = applyCARules(updatedGrid, birthThreshold, survivalThreshold,
calculate_transition_probabilities(soil_mixture_index, low_land_value, land_surface_temperature,
geological_floodplain, distance_to_street));
end

imagesc(updatedGrid);
title('Final Urban Density Grid');
colormap(gray);
axis square;

% Extract relevant population data for the specified years (1977-2022)
Var1_1977_2022 = 1977:2022;
population_data_1977_2022 = population_data(ismember(population_data.Var1, Var1_1977_2022), :);

% Calculate population transitions
transitions = diff(population_data_1977_2022.Var2);
```

```
% Compute transition probabilities matrix
transition_probabilities_matrix = transition_matrix(transitions);
disp('Markov Transition Probabilities Matrix (1977-2022):');
disp(transition_probabilities_matrix);

figure;
subplot(2, 3, 1);
imshow(soil_mixture_index);
title('Soil Mixture Index');

subplot(2, 3, 2);
imshow(low_land_value);
title('Low Land Value');

subplot(2, 3, 3);
imshow(land_surface_temperature);
title('Land Surface Temperature');

subplot(2, 3, 4);
imshow(geological_floodplain);
title('Geological Floodplain');

subplot(2, 3, 5);
imshow(distance_to_street);
title('Distance to Street');

sgtitle('Suitability Maps');

years_2020_2040 = [2020, 2030, 2040];
for year = years_2020_2040
    figure;
    projected_map = simulate_CA_Markov(initialGrid, numIterations, birthThreshold, survivalThreshold,
calculate_transition_probabilities(soil_mixture_index, low_land_value, land_surface_temperature,
geological_floodplain, distance_to_street), year);
    imagesc(projected_map);
    title(sprintf('CA-Markov Projected Land-Use and Land-Cover Map for %d', year));
    colormap(gray);
    axis square;
end

figure;
subplot(2, 3, 1);
imagesc(soil_mixture_index);
title('Soil Mixture Index');
colormap(gray);
axis square;

subplot(2, 3, 2);
imagesc(low_land_value);
title('Low Land Value');
colormap(gray);
axis square;

subplot(2, 3, 3);
```

```
imagesc(land_surface_temperature);
title('Land Surface Temperature');
colormap(gray);
axis square;

subplot(2, 3, 4);
imagesc(geological_floodplain);
title('Geological Floodplain');
colormap(gray);
axis square;

subplot(2, 3, 5);
imagesc(distance_to_street);
title('Distance to Street');
colormap(gray);
axis square;

sgtitle('Criterion Raster Maps');

function updatedGrid = applyCARules(initialGrid, birthThreshold, survivalThreshold, transition_probabilities)
  [rows, cols] = size(initialGrid);
  updatedGrid = zeros(rows, cols);

  for i = 1:rows
    for j = 1:cols
      neighbors = sum(sum(initialGrid(max(1, i-1):min(rows, i+1), max(1, j-1):min(cols, j+1))));
      currentCell = initialGrid(i, j);

      % Calculate transition probability based on suitability maps
      transition_prob = transition_probabilities(i, j);

      % Apply transition rule based on birthThreshold, survivalThreshold, and transition_prob
      if currentCell == 0 && neighbors == birthThreshold && rand() < transition_prob
        updatedGrid(i, j) = 1;
      elseif currentCell == 1 && (neighbors == survivalThreshold || neighbors == survivalThreshold + 1)
        updatedGrid(i, j) = 1;
      end
    end
  end
end

function updatedGrid = applyDensityConstraints(initialGrid, minDensity, maxDensity)
  currentDensity = sum(initialGrid, 'all') / numel(initialGrid);

  if currentDensity < minDensity
    numToAdd = round((minDensity - currentDensity) * numel(initialGrid));
    randomIndices = randperm(numel(initialGrid), numToAdd);
    initialGrid(randomIndices) = 1;
  elseif currentDensity > maxDensity
    numToRemove = round((currentDensity - maxDensity) * numel(initialGrid));
    randomIndices = randperm(numel(initialGrid), numToRemove);
    initialGrid(randomIndices) = 0;
  end
```

```
    updatedGrid = initialGrid;
end

function transition_probabilities = calculate_transition_probabilities(soil_mixture_index, low_land_value,
land_surface_temperature, geological_floodplain, distance_to_street)
    % Ensure all suitability maps have the same dimensions
    min_rows = min([size(soil_mixture_index, 1), size(low_land_value, 1), size(land_surface_temperature, 1),
size(geological_floodplain, 1), size(distance_to_street, 1)]);
    min_cols = min([size(soil_mixture_index, 2), size(low_land_value, 2), size(land_surface_temperature, 2),
size(geological_floodplain, 2), size(distance_to_street, 2)]);

    soil_mixture_index = soil_mixture_index(1:min_rows, 1:min_cols);
    low_land_value = low_land_value(1:min_rows, 1:min_cols);
    land_surface_temperature = land_surface_temperature(1:min_rows, 1:min_cols);
    geological_floodplain = geological_floodplain(1:min_rows, 1:min_cols);
    distance_to_street = distance_to_street(1:min_rows, 1:min_cols);

    % Normalize suitability maps
    soil_mixture_index = double(soil_mixture_index) / 255;
    low_land_value = double(low_land_value) / 255;
    land_surface_temperature = double(land_surface_temperature) / 255;
    geological_floodplain = double(geological_floodplain) / 255;
    distance_to_street = double(distance_to_street) / 255;

    % Calculate transition probabilities
    transition_probabilities = 0.2 * soil_mixture_index + 0.1 * low_land_value + 0.3 * land_surface_temperature + 0.2 *
geological_floodplain + 0.2 * distance_to_street;
end

function transition_probabilities_matrix = transition_matrix(transitions)
    % Compute transition counts
    unique_transitions = unique(transitions);
    num_unique = length(unique_transitions);
    transition_counts = zeros(1, num_unique);
    for i = 1:num_unique
        transition_counts(i) = sum(transitions == unique_transitions(i));
    end

    % Compute transition probabilities matrix
    transition_probabilities_matrix = zeros(num_unique);
    for i = 1:num_unique
        from_state = unique_transitions(i);
        transitions_from_state = transitions(transitions == from_state);
        unique_transitions_to = unique(transitions_from_state);
        num_unique_to = length(unique_transitions_to);
        for j = 1:num_unique_to
            to_state = unique_transitions_to(j);
            count = sum(transitions_from_state == to_state);
            idx_to = find(unique_transitions == to_state);
            transition_probabilities_matrix(i, idx_to) = count / transition_counts(i);
        end
    end
end
```

```
function projected_map = simulate_CA_Markov(initialGrid, numIterations, birthThreshold, survivalThreshold,
transition_probabilities, Var1)
  updatedGrid = initialGrid;
  for iteration = 1:numIterations
    updatedGrid = applyCARules(updatedGrid, birthThreshold, survivalThreshold, transition_probabilities);
  end
  projected_map = updatedGrid;
```