International Journal of Informatics Media and Communication Technology
ISSN: Online 2682-2881 & Print 2682-2105

SLEUTH-CNNs Model: A Proposed Conceptual Modification of a Cellular Automata based Geo-Artificial Intelligence Modeling Multi-System

Nermin A. Shoukry
Department of Geography, Faculty of Arts, Cairo University, Giza, Egypt.

Submit Date: ***19/12/2024*** **Accept Date:** ***23/12/2024***

DOI: https://doi.org/10.21608/ijimct.2024.345918.1072

Abstract

The cellular automata-based SLEUTH is an independent fully operated intelligent dynamic simulation and prediction model for urban growth. It was released simultaneously with the beginning of the second millennium by the geographical scientist "Keith Clarke" and his colleagues in the U.S.A. This model infused computer science and programming into the science of urban geography. Although it was first designed to simulate and predict an American urban environment, it was universally acclaimed by many researchers around the world. Thus, many different applications of the model spread across the globe, resulting in hundreds of published scientific papers that used it world widely. Yet, the geographical environments are varied from study area to another in most researches and its final output results were mostly insignificant and not reflecting the reality. Being an open-source code model, SLEUTH has gone through several evolutionary stages of programmatic modifications either were done by few modelers or by Clarke himself with his co-modelers. The most challenging and critical modeling modification is concerning the improvement of the algorithm's performance in terms of calibration and reducing computation time while maintaining its basic structure. The SLEUTH-GA model is one modification of the model that aims to replace the Brute Force (BF) calibration method with a Machine Learning based Genetic Algorithms (GA). From this respect, this research is a new attempt to offer a novel modification methodology for creating a complete modeling integrated multi-system which consists of a set of artificial intelligence subsystems that are implemented together in a parallel programming approach. This proposed modification transformed SLEUTH from being an applicable intelligent model to be an individual complete intelligent software program. The novelty of this research also comes in proposing a full integration of the Deep Learning (DL) based Convolutional Neural Networks (CNNs) in both processing and calibration modeling modes. This promising attempt of this suggested modification offers a new Geo-Artificial Intelligence approach that can be used by many researchers not only to simulate and predict future urban growth, but also it can be applied for other geographical growing phenomena.

Keywords: *GeoAI, Cellular Automata (CA), SLEUTH, UGM, Machine Learning (ML), Genetic Algorithms (GA), Deep Learning (DL), Convolutional Neural Networks (CNNs).*

نموذج SLEUTH-CNNs: تعديل مفاهيمي مقترح لنموذج ذكاء إصطناعي جغرافي متعدد الأنظمة قائم على أساس الأوتوماتا الخلوية

نرمين أحمد شكري

قسم الجغرافيا–كلية الآداب–جامعة القاهرة

ملخص البحث:

يعد نموذج SLEUTH القائم على الأتمتة الخلوية نموذجا ذكيا ومستقلا للمحاكاة الديناميكية والتوقع المستقبلي للنمو الحضري. وقد تم إصداره تقريبا مع بداية الألفية الثانية من قبل العالم الجغرافي "كيث كلارك" وزملائه في الولايات المتحدة الأمريكية. حيث أنه ولأول مرة في تاريخ علم الجغرافيا، دمج هذا النموذج علوم الكمبيوتر والبرمجة في علم الجغرافيا الحضرية. وعلى الرغم من أنه تم تصميمه في البداية لمحاكاة وتوقع البيئة الحضرية الأمريكية، إلا أنه قد حظي بإشادة عالمية من قبل العديد من الباحثين في جميع أنحاء العالم. وبالتالي، انتشرت العديد من التطبيقات المختلفة للنموذج في جميع أنحاء العالم، مما أدى إلى نشر مئات الأوراق العلمية التي استخدمته على نطاق واسع في جميع أنحاء العالم. ومع ذلك، فإن البيئات الجغرافية تختلف من منطقة دراسة إلى أخرى في معظم الأبحاث وكانت نتائج مخرجاتها النهائية غير صحيحة في الغالب ولا تعكس الواقع الجغرافى الحقيقي. ولأنه نموذج مفتوح المصدر برمجيا، فقد مر SLEUTH بمراحل تطورية عديدة من التعديلات البرمجية التي قام بها إما عدد قليل من المصممين أو من قبل كلارك نفسه مع المصممين المشاركين له. ويعد التعديل الأكثر تحديًا وأهمية في النمذجة يتعلق بتحسين أداء الخوارزمية من حيث المعايرة وتقليص زمن أداء العمليات الحسابية البرمجية مع الحفاظ على بنيتها الأساسية. و يعد نموذج SLEUTH-GA تعديلا من تعديلات النموذج الذي يهدف إلى استبدال أسلوب

معايرة القوة الغاشمة (BF) بالخوارزميات الجينية (GA) القائمة على أساس منهج التعلم الآلي داخل نظام المعايرة النمذجية. ومن هذا المنظور، يعد هذا البحث محاولة جديدة لتقديم منهجية تعديل جديدة ومستحدثة لإنشاء نظام متعدد متكامل للنمذجة يتكون من مجموعة من أنظمة الذكاء الإصطناعي الفرعية التي يتم تنفيذها معًا في نهج البرمجة المتوازية. قام هذا التعديل المقترح بتحويل SLEUTH من كونه نموذجًا ذكيًا قابلاً للتطبيق ليتحول كلية إلى برنامج حاسوبي ذكي ومستقل. تأتي حداثة هذا البحث أيضًا في اقتراح التكامل الشامل للشبكات العصبية الإلتفافية (CNNs) القائمة على أساس التعلم العميق (DL) في كل من نظامي المعالجة والمعايرة النمذجية. وأخيرا تقدم محاولة التعديل المقترحة الواعدة هذه نهجًا جديدًا للذكاء الإصطناعي الجغرافى GeoAI يمكن للعديد من الباحثين استخدامه ليس فقط لمحاكاة وتوقع النمو الحضري المستقبلي، ولكن يمكن أيضًا تطبيقه على كل ظاهرات النمو الجغرافية الأخرى.

الكلمات المفتاحية: SLEUTH، UGM، التعلم الآلي (ML)، الخوارزميات الجينية (GA)، التعلم العميق (DL)، الشبكات العصبية الإلتفافية (CNNs).

I. Introduction

The unconquered mind of human being led to have an uncountable number of innovations that serve the humanity in various scientific fields. Artificial intelligence is considered as one main piece of jewelry among the crown jewels of the modern human creation ingenuity. Thus, following the rise of the "*GIScience-GeoComputation Revolution*" period in the late 20th century, different intelligent approaches have been attempted to be practiced in many geospatial scientific disciplines; especially in the field of geography. From various geospatial applications, two specific and crucial applicable geospatial topics such as "settlement/urban growth" and "land use/land cover (LULC) change" were enlightened and examined by some researchers. For this reason and out of scientific necessity, many innovative intelligent modeling approaches were created to simulate and predict the complexities of these two connected geographical phenomena. Various grid-based Geo-Artificial Intelligence modeling algorithms enable us to simulate and predict the complexities of the growth process. For example, Urban growth intelligent dynamic models such as Land Change Modeling (LCM), System Dynamics (SD), Cellular Automata (CA), Agent based, Fractal, Fuzzy Modeling, Genetic Algorithms (GA) and optimization, Bayesian Network (BN), Multi-agent systems, Swarm Intelligence, Markov Chain (MC), Land Transformation Model (LTM), CA-SLUETH, Artificial Bee Colony algorithm (ABC), and Artificial Neural Networks (ANN) modeling have shown great capability for representing, simulating, and forecasting the complexity of urban growth change system. Additionally, the past decade confronts many hybrid and integrated intelligent programmed algorithms that were created to improve prediction performance by integrating various modeling techniques such as: "ABC-CA urbanization model", "SD and CA", "CA and

BN", "CA and MC", and "ANN and CA" (Li & Yeh, 2002; Kocabas & Dragicevic, 2007; Guan *et al*., 2011; Xu *et al*., 2014; Feng & Batty, 2016; Naghibi *et al*., 2016; Zare *et al*., 2017). Most recently, an effective new born computer science approach of incorporating hybrid intelligent algorithms of cellular Automata that has embraced deep learning (DL) -it is a type of machine learning method- whose structured all Neural Networks (NN) approaches. It has been created as one independent intelligent system, but unfortunately, it is not yet spread in the geographical and geospatial sciences (till the publishing date of this research study). It is often referred to as Neural Cellular Automata (NCA) modeling approach (Najarro *et al*., 2022; Tesfaldet *et al*., 2022).

In this respect, two main objectives of this research are meant to achieve. Firstly, is to focus on enlighten and conceptually modified an intelligent Cellular Automata Modeling Approach such as SLEUTH, and secondly to propose a new Intelligent Information Multi-System. Regarding The CA-based Geospatial Artificial Intelligence (GeoAI) modeling, it was conceptually and programmatically practiced by many modelers, each had his own modeling programming language and scientific applicable purpose. Some modelers developed and designed their CA models as independent programmed stand-alone algorithm that could not be processed or executed except in the previously selected and used programming language software. While, some other CA models that despite the fact of being also independently programmed, it would be designed to be capable of being linked to other software. Taking the credit, [*] Keith Charles Clarke (Clarke, K.C.) is a pioneer model with its enormous applications worldwide and its evolution over 27 years that starts from its first appearance till current publishing date of this research. Followed by highlighting the technical modifications attempts that

[*] Professor of Analytical Cartography and Modeling in the department of Geography-The University of California-Santa Barbara (UCSB), U.S.A.

geographer in dynamic simulation and prediction of CA-based computer modeling. In the year of 1997, he created the first independent fully operated "Urban Cellular Automata" programmed intelligent model (Clarke, *et al*., 1997). One year later, Clarke and Gaydos innovated the linkages between CA model and GIS environment (Clarke & Gaydos, 1998). Structurally, this research highlights the initial notion of CA modeling and explains its detailed characteristics, structure, properties, and mechanism. Moreover, it provides a full description of GeoComputational based "SLEUTH CA"

II. Literature Review

The literature review of this research provides the necessary background of the variation of the created and developed grid-based CA urban growth dynamic (simulation and prediction) models. Moreover, it offers a basic understanding of the concepts, mechanisms, characteristics, properties, and capabilities of CA model as a Geo-Artificial Intelligence dynamic modeling foundation. Moreover, it is directed toward explaining the fundamentals of SLEUTH-UGM Model, as well as discussing its evolution and the various modification attempts to enhance its

---

were done by its original creator with his co-modelers, or by some other local (U.S.A) or global (worldwide) researchers. These modifications are mainly focused on modifying calibration mode and computational performance. For example: creating a new optimal set of metrics, improving the computational performance, creating a regional version of the model, redesigning calibration by adding choices on encoding or changing to Machine Learning based Genetic Algorithms, enhancing optimization of transition rules, executing and managing subsets of the parameter's space on a shared memory that decrease calibration time, and reducing computational load with sufficient execution memory (Dietzel & Clarke, 2008; Guan & Clarke, 2010; Jantz, *et al*., 2010; Clarke-Lauer & Clarke, 2011; Chandan, *et al*., 2019; Chaudhuri & Foley, 2019; Clarke & Johnson, 2020).

performance over time. Therefore, each related literature review will be embedded and demonstrated sequentially in each following subjects of this research paper.

III. Research Overall Methodology Structure

The overall research methodology is displayed in Figure 1. It is divided structurally into four main parts. The first part is discussing the significant grid-based fully operated dynamic CA models during the time period of (1987-2019). The second part is focusing on displaying on details the components, characteristics, properties, and mechanism of CA-based GeoAI modeling. Whereas, third part examining traditional SLEUTH model with its characteristics and structure. Moreover, it compares it operationally with all modification modeling attempts. Finally, the last part of the research is functionally explaining the proposed "SLEUTH-CNNs" urban growth model and its two main modes. The first mode is the "data preparation & processing" that experienced a novel merging of Deep Learning-Convolutional Neural Networks (DL-CNNs) as one major suggested approach of data processing. Regarding the second mode of "model calibration & prediction", its calibration technique's novelty is to suggest a fully integration that would be able to couple both "Deep Learning based CNNs" and "Machine Learning based Genetic Algorithms" together in a parallel operational function. The final research output is the resultant fully designed intelligent dynamic modeling multi-system.

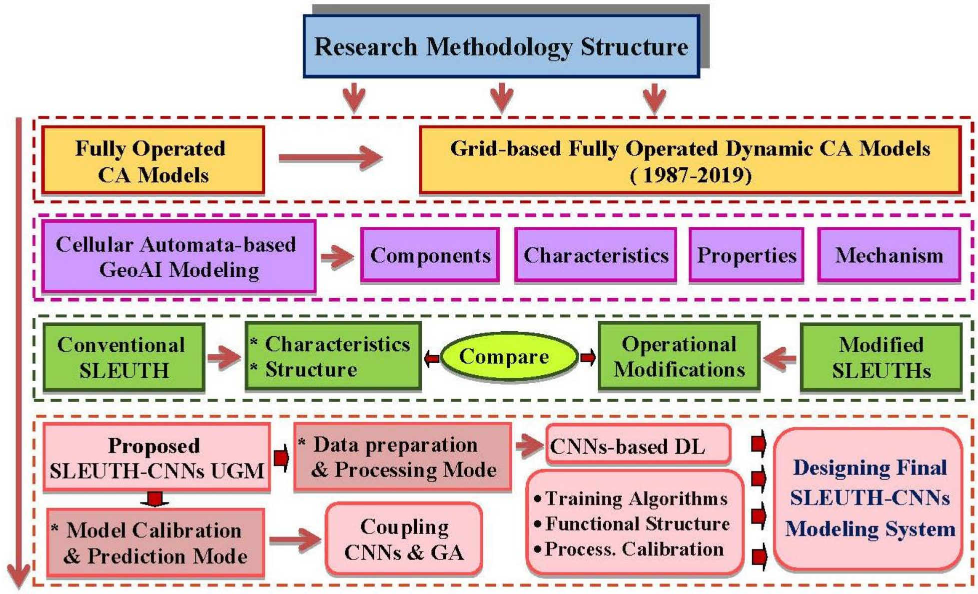

Figure 1. Research Schematic Flowchart showing Methodology Structure.

IV. Grid-based Dynamic CA Fully-Operated Urban Models

The notion of creating an urban dynamic simulation and prediction modeling was and still a hopeful scientific target that many researchers seek to reach and accomplish. Michael Batty was a pioneer British multi-discipline specialized geographer, whose research works concern with the development of data systems and computer models of cities and regions. In 1971, Batty published his famous research study of "Modeling Cities as Dynamic Systems". He was inspired mainly by the works of two other pioneers and innovators American scientists. The first was the economist and statistician "John Patrick Crecine", who was contributed to create a statistical dynamic model of urban structure (Crecine, 1968). The second was the scientist "Jay Wright Forrester" who is specialized in computer systems, and also was the founder of computer system dynamics that was established as an academic discipline. In addition, Forrester was the creator of the well-known "urban dynamics model" in the late sixties of the 20th century (Forrester, 1969). Batty in his research study designed an initial dynamic spatial statistical model to simulate the changing urban structure and activities over a period of 15 years (Batty, 1971).

Afterwards, urban simulation and prediction dynamic models especially the CA-based models; they became more advanced and sophisticated with the frequent evolution of computer science and programming technologies. Most recently, these cellular automata based urban models have incorporated several additional GeoAI methods to CA. Among all urban simulative and predictive CA models, there are some suggested models that are fully operated either as independent computer programmed experimental models or as models that later were evolutionary transformed to be an independent stand-alone software package. Moreover, some of these models were created academically as empirical software's program applications with no commercial profit, while others were originally created as profitable commercial software packages. Operationally functioned, there are some stand-alone CA-based GeoAI modeling software packages that were released to be "ready or semi-ready to use" modeling software to simulate and predict future urban spatial dynamics. This research monitored 13 important stand-alone CA-based software packages for raster (grid-based) urban dynamic modeling systems, such as: LCM-TerrSet (IDRISI), Metronamica, (Clue, Clue-s, and Dyna-clue), SLEUTH, (Dinamica EGO, DinamicaVM), SIMLANDER, APoLuS, LUCSIM, GeoSOS Flus, and UrbanCA (Figure 2).

Land Change Modeler (LCM) software module is developed by Clark Labs, which is a non-profit laboratory. It is an integrated system of both GIS-geospatial analytical system with remotely sensed image processing system to evaluate, process, analyze, simulate and predict future land cover change (Eastman, 2016b). Even though LCM module was released in the year of 2009, it is originally conceived from IDRISI GIS software that was created by the geographer "J. Ronald Eastman" and his team work in Clark University (USA) in 1987 (Eastman, 2016a; Eastman & Toledano, 2018). Starting from 2015, an IDRISI version was released and transformed its name with the name of "TerrSet" that contained eight modules and it is focused on incorporating raster-based GIS analyses with image processing and supported by a programming language (Zhou & Hu, 2016). Up to 2023, the evolution of LCM's programming languages (PLs) were upgraded and redeveloped from C++, to Python, and to Delphi as well as upgrading its programming-based methods such as: Markov Chain, Multi Objective Land Allocation, and finally Artificial Neural Networks (Sipahioglu & Cagdas, 2023).

METRONAMICA is a constrained CA dynamic software package that is carried out modeling the land use/cover change (LUCC). Originally, it is based on the conceptual theories that are offered by Couclelis in 1985, White & Engelen in (1993, 1997), and White and his fellow researchers (Engelen & Uljee) in 1997. It is created and developed in 1992 by the Research Institute for Knowledge Systems (RIKS BV) in the Netherlands. This modeling system is developed in the GEONAMICA software programming environment. It is a commercially paid and stand-alone software application that is compatible with standard GIS packages such as ArcGIS (RIKS BV, 1992). It has the capacity to model a wide range of urban land uses exceeded 26 classes and allows setting the parameters values interactively with exploring the model behavior visually and in real time (Stanilov & Batty, 2011). The method that is supporting CA in this software package is NASZ (Neighborhood-Accessibility-Suitability-Zoning).

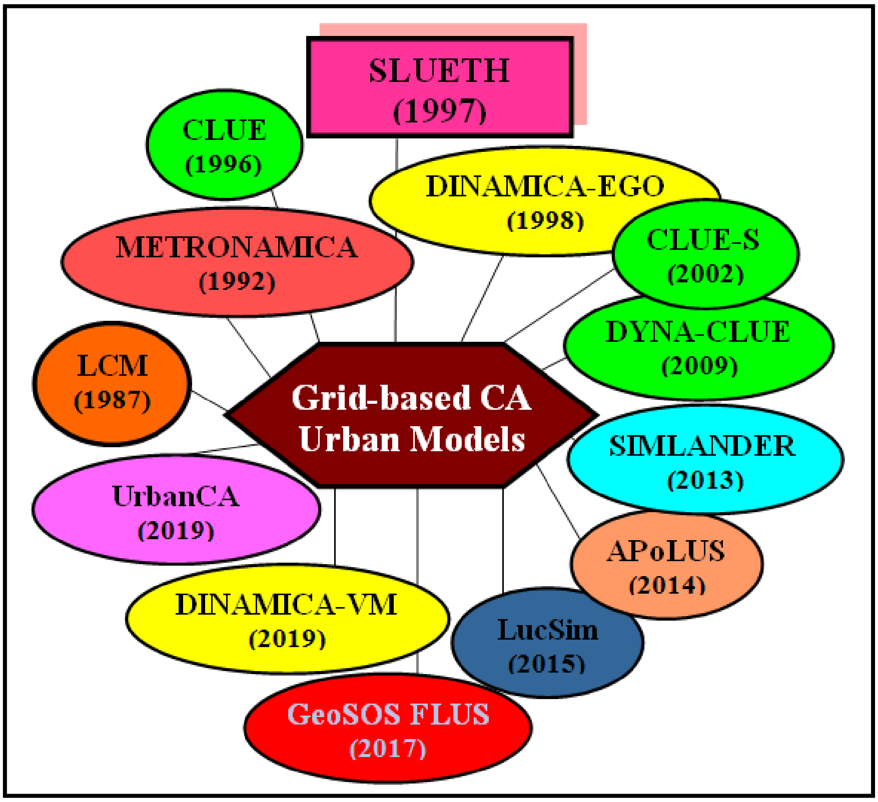

Figure 2. Grid-based Dynamic CA Fully-Operated Urban Simulation & Prediction Models in the period of (1987-2019).

CLUE (Conversion of Land Use and its Effects) is an individualized dynamic modeling software package that is created by Veldkamp & Fresco in 1996. It is developed for national to continental level to simulate LUCC using empirically quantified relations between each land use and its driving factors and associated with dynamic modeling. It is created by using PASCAL software programming language (PL) and supported by CA-based Multi Objective Land Allocation (MOLA) method. It is followed by a sequence of upgraded modified software package such as: CLUE-S (for small and local regional extent) (Verburg, et al., 2002) Dyna-CLUE (programmed in "DYNA"/ using MOLA method) which examines the implications of different scenarios of the combination of top-down and bottom-up processes within a single allocation algorithm (Verburg & Overmars, 2009).

SLEUTH is the main modeling modification's objective in this research. It is the widely popular and used CA-based dynamic urban growth model. It is mainly simulating and predicting the growth by simply using a binary distinction between urban and non-urban land cover. It is also an open-source software package and easy to improve its source code. It was created by using (C) software programming language (PL) under UNIX, and supported by a constraint Cellular Automata method (Clarke, Hoppen, & Gaydos, 1997). Since its first release in 1997, many researchers in USA and other worldwide developed and developing countries have used its primitive first version to simulate and predict the growth of their selective urban areas. One pioneer application study outside USA and Europe was examined and achieved in Alexandria of Egypt in 2004. It is visually predicted the Alexandria's urban growth trends for almost 50 years ahead by using different source of moderate spatial resolution datasets. The prediction process used only one scenario of stabilizing the initially measured growth rate. Despite the fact of having several obstacle factors such as: data limitation especially for roads and slope, unavailable higher spatial resolution datasets, and limitations of computational power that leads to long run times; the final result was reasonable and succeeded to predict the future general urban growth trends of the study area (Azaz, 2004). Years later, due to the limitation of computational functionality and other calibrated built-in controlling coefficient factors of the "ready-to-use" dynamic CA model, SLEUTH was modified frequently by several individual researchers, by Clarke and his fellow modelers, and also by governmental organizations. Among further modifications, is the one was done by "Claire Jantz" with his research colleagues as cooperation research of "United States of Geological Survey

(USGS) & The Eastern Geographic Science Center (EGSC)" (Jantz, *et al*., 2010; www.sciencebase.gov, 2019). The latest available open-source software version according to "Project Gigalopolis-University of California/Santa Barbara (UCSB)" is the (SLEUTH-3r) which is designed to be a regional modeling version (Gigalopolis, 2003).

DINAMICA EGO is created and released in 1998 by Britaldo Silveira Soares-Filho the Brazilian cartographer. It is recently applied for urban growth simulation modeling, yet it was focused more on environmental changes modeling up to the year of 2015. It was created by using (C++) programming language software, then upgraded to be coupled with (Java) and supported by a Multi Criteria Evaluation based Cellular Automata method. Moreover, it was evolved in a version called Dinamica virtual machine (DinamicaVM) that is designed to overcome some shortages that appeared in the first version and to make its framework more flexible. In this version the programming environment was implemented in Java, while the instruction set architecture and runtime environment were implemented in C++ (Ferreira, *et al*., 2019).

SIMLANDER is a designed CA land use model especially for the R software programming language environment. It is released in 2013 and created by R. Hewitt, J. Pacheco, & B. Gomez (Hewitt, *et al*., 2013). In 2022, Hewitt and his associated research modelers offered an upgrading version of SIMLANDER modeling framework with a proposed multi-model ensemble approach to build eight different CA models for the R environment (Hewitt, *et al*., 2022).

APoLUS (Actor, Policy and Land Use Simulator) is a model that links together three modeling approach such as: the spatially explicit geographical model, the policy implementation theory, and the sociological approach. It was initially known as PLUS4-CMP, but it was renamed to its current acronym in 2014. Furthermore, it was created and developed from SIMLANDER by Richard Hewitt to include multiple cell states and additional planning decisions. It was also implemented in R and adopted the NASZ cellular automata method (Hewitt, 2015).

LucSim (Land use change Simulation) is an experimental model build to understand, simulate, and forecast land use changes (LUC). Its principal functionality is based upon the major definitions that were set by Paul Torrens (Torrens, 2011). In 2015, it was designed and developed by Antoni & Vuidel at ThéMA laboratory in France. It is based on Cellular Automata-Markov Chain method and implemented and running by Java 8 programming software environment. Moreover, it is a user-friendly software that is adapted for analyzing and simulating LUC and spatial dynamics at different scales. In this modeling approach, urban CA incorporates spatial and non-spatial dynamics as well as heuristic (Antoni, *et al*., 2019).

GeoSOS FLUS (Geographical Simulation and Optimization System -Future Land Use Simulation). It is developed by the geographer Xiaoping Liu and eight other associate researcher-mates in 2017. It is an integrated model to deal with multi-type land use scenario simulations by coupling both human and natural effects. Therefore, it is established by coupling both intelligent cellular automata (CA) and artificial neural networks (ANNs) based urban simulation models. It is similar to LCM, yet its additional strength is the infusion of ANNs. As well as it is free and user-friendly software. It is totally created by using C++ software programming language and supported by both compiling CA-based Multi Objective Land Allocation (MOLA) and ANNs intelligent methods. Furthermore, it allows users to simulate different designed land use and land cover change (LUCC) scenarios with multiple cell states (Liu, *et al*., 2017).

UrbanCA is a spatially explicit CA modeling framework free software package that is simulating urban growth dynamics and projecting future scenarios across scales. It is experimentally developed and released by Feng and Tong in 2019. It is based on reconstructing the essential CA structure and incorporating non-spatial {Logistic Regression (LR) method, and Generalized Additive Model (GAM)}, spatial {Spatial Auto Regressive (SAR), Geographically Weighted Regression (GWR)}, and heuristic methods (Optimization algorithms to calibrate CA transition rules). This UrbanCA framework Contains four main parts: basic technique architecture, transition rule modeling, model implementation, and model accuracy assessment (Feng & Tong, 2019).

V. Cellular Automata based GeoAI Modeling

Raster or grid based Cellular Automata Model Sometimes Automaton (CA) were first developed in the early post-World War II. It specifically introduced in 1948 by Von Neumann and Ulam to model complex dynamic systems of self-reproducing automata (Burks, 1970). Later, Ulam and Schrandt realized the self-reproducing automata with simpler models than those used by Von Neumann. These CA systems have had an active life (Beyer *et al.*, 1985). A famous pioneer applied computer-based example of a CA is "The Game of Life", developed by John Conway in 1970. He exposed these CA ideas through other fields of science. Simple simulations of death and life of cells in a CA game proved the striking similarities between real life and simulated life in a computer (Gardner, 1970).

Afterwards, CA have attracted the attention of researchers in geography because of its ability to model and visualize complex spatially distributed processes that can model two-dimensional space. Based on this feature, some geographers have adopted CA to simulate land uses and other geographical phenomena (White &

Engelen, 2000). In a pioneering paper, the geographer Waldo Tobler was the first to consider using explicit cellular automata for modeling geographical phenomena. He describes the theoretical foundation for a Cellular Geography and examines all formal possibilities of cell transitions according to different processes involving their neighborhood by using a land use modeling as an experimentally geographic example. (Tobler, 1979).

Another geographer – Helen Couclelis – influenced by Tobler, and continued exploring the way in which strict CA might be applied to geographical systems. Additionally, she provides theoretical and practical frameworks for using CA in modeling the dynamics of geographical spatial phenomena as well as modifying and improving Tobler's work (Couclelis, 1985, 1997). However, not until the late 1980s and early 1990s, the thinking of using this technique is really began to rise simultaneously with the flourish of computer programming, graphics, and complexity which all together help to generate most conditions that were shedding more lights on the possibility to study different geographical applications.

Toffoli and Margolus (1987) proposed CA models to replace differential equation-based models. The CA were used afterward to model process-based surface water quality and fire propagation problems. It grew out of earlier environmental simulation work on the wildfire behavior.

White and Engelen (1993, 1997) integrated models for socio-economic and natural systems to predict the future demand for the various land uses. Later, they have developed several CA and CA-based integrated models that are designed as prototypes of spatial decision-support systems for urban and regional planning and impact analysis. They proved that their CA models are effective at generating realistic simulations of both land use patterns and other spatial structures (White & Engelen, 2000).

Takeyama & Couclelis (1997) demonstrated the role of CA as potential powerful contributions to urban process modeling. They presented in their research paper the modeling formalism of CA that is generalized and extended within their proposed approach, called Geo-Algebra which is considered as an extension and new generalization of Tomlin's Map Algebra. They described the integration of CA with GIS and they named their method "Map Dynamics". They concluded that it can allow the modeling of additional dynamic behaviors. Finally, they argued that their proposed approach will not only make CA more widely applicable, but will also significantly enhance the general modeling capabilities of GIS technology itself.

Afterwards, Batty, Xie, and Sun (1999) developed the Dynamic Urban Evolutionary Model (DUEM) to model land use change through different transition rules. They specified and defined various decision rules that embedded distance and direction, density thresholds, and transition or mutation probabilities into the model's dynamics.

Over the last two decades, there are steadily increasing studies on simulating urban growth using CA techniques. It shed lights on the application of CA in urban modeling that can give insights into a wide variety of urban phenomena. Most studies argued that urban CA models have better performance in simulating urban growth than conventional urban models because they are much simpler than complex mathematical equations and producing results that are more meaningful and useful. Moreover, most scientific papers enhanced the theoretical and methodological aspects of CA for analyzing and modeling complex urban dynamics on one hand. On the other hand, its emphasized on the concept of the temporal and spatial complexities of urban systems and this can be well modeled by properly defining transition rules in CA models (Deadman, *et* *al*., 1993; White & Engelen, 1993, 1997, 2000; Batty & Xie, 1994; Wu, 1998; Batty *et al*., 1999; and Li & Yeh, 2002).

In 1994, another successful geographical attempts to apply CA model, was proposed by Keith Clarke and his team work colleagues to predict the spatial and temporal behavior of wildfires (Clarke, *et al*., 1994). Following his sequential research papers, Clarke expands his interest of CA model and developed the Urban Growth Model (UGM) based on integrating both GIS and CA approaches. The UGM simulates the urban growth transition from non-urban to urban land (Clarke, *et al*., 1997). In this model, there are two main types of factors that are treated as modeling input. The first type is the local factor that consists of several sub-factors such as roads, existing urban area, and slope. The second type is the temporal factor, which conducts the historical patterns of growth. The simulation is controlled by five parameters, which carry respective weights or coefficients: slope resistance, road gravity, breed, dispersion, and spread. The coefficient of each parameter is determined by running four calibration phases: coarse, fine, final, and averaging best results. The weighted probabilities of each parameter are then used as input into the growth prediction phase. Afterwards, Clarke, Gaydos, and Hoppen coupled two CA models -UGM (Urban Growth Model) and LUCM (Land Use Change Model) - to predict urban growth as a part of a project for estimating the regional and broader impact of urbanization. The first conduct of the model is to apply their basic UGM using weighted maps as inputs and applied a set of CA rules that determined whether or not cells will change from non-urban to urban for each year in sequence in their study area in the San Francisco Bay area. This was coupled to a second CA model that received the number of new urban cells in each time period to create and enable change among land uses other than the urban class, for example: changing from wetlands to agriculture (Clarke, *et al*., 1997). As a result, they produced their first fully operated CA urban model as it was mentioned previously in the above text.

Evolving their research works, Clarke and Gaydos developed further CA model that loosely coupled with grid-GIS to conduct not only the description, simulation, visualizing, and modeling the urban growth, but also to predict future urban transition. Concerning the part of their CA urban growth simulation model, they applied trial and error approach as one major type of their calibration methods by comparing their output simulation results visually using different combinations of parameter values for their various study areas in different geographical environments in the U.S.A, such as San Francisco Bay in California and the Washington DC/Baltimore Corridor (Clarke & Gaydos, 1998). Thus, this cell-based urban growth/urban land use change model that was produced by Keith Clarke and his fellow researcher, has been introduced to the (GeoComputational/GeoScience/GeoModeling) Scientific Society with the name of "The CLARKE CA-UGM" model. After technical modifications, it was named as " SLEUTH Model", which is an evolution of the formerly UGM. It is a CA-based urban growth simulation model that coupled with a land use/cover change "LUCC" model. The SLEUTH initials referred to the five main categories of data inputs. It stands for: **S**lope, **L**and cover, **E**xclusion area, **U**rban extent, **T**ransportation network, and **H**illshade. Whereas the UGM is designed for local applications, the SLEUTH is designed to be more ambitious and claimed to be used for forecasting and predicting urban growth at a regional and even continental scale. The model's programming source code is written in C programming language, and support three different modes: Test, Calibration, and Prediction. This CA automated simulation model highly depends on using many historical land use and other data to project growth and land use change into the future. One main reason behind using this time series historical data, is to derive the behavioral parameters that best capture the structure and dynamics of the location-specific growth history. World widely, the SLEUTH model has been used in an enormous number of studies that has been proven to be suitable for modeling the future urban expansion of Metropolitan, Megalopolis, and even Gigalopolis areas. It has been among the leading applied cellular automaton (CA) models that was used to simulate land use/cover change (LUCC) at many different spatial scales.

Moreover, it is during the last two decades that extensively going toward increasing the use of CA models for urban growth simulation, can be detected as a growing field of interest among geographers. As mentioned above, many researchers applied and modified UGM-SLEUTH model all over many geographical cities and regions of the world including Keith Clarke himself. The model is used mainly for verifying urban growth or used as a scenario-based growth dynamics and future predictions. Despite the promising and remarkable achievements of the SLEUTH model, its performance can be further improved in many aspects considering the processing complexity, the ability, the reliability, and the calibration methods concerning the task of the modeling computational speed.

Recently, some advanced applications exceeded the use of a stand-alone CA model and transformed from being able to operate and function independently and is not incorporated into other existing systems, to be able to merge and incorporate into other systems such as: GIS, remote sensing, and the paradigm of the fuzzy-set based on the fuzzy logic theory with cellular automaton approach (Liu & Phinn, 2003). As well as coupling CA model with some other advanced intelligent models such as Artificial Neural Networks (ANN) to overcome its deficiency of specific parameters and applications (Li & Yeh, 2002, 2004). Evolutionary, the next-generation of urban modeling needs to conceptualize the dynamics of the world's megacities, which are, in many instances, growing in number, size, and influence at unprecedented rates (Torrens, 2012). This will lead to develop many hybrids, coupled, or independent stand-alone intelligent mega multi-system models to anticipate simulations and future growth predictions of the world's megacities.

1. *Understanding Cellular Automata Modeling*

A cellular automaton is a discrete dynamical system which is created to model complex natural systems that are examined in computability theory, mathematics, physics, and theoretical & microstructure modeling. The space, time, and states of the system are discrete (Hurd, 1987). In its simple definition, the CA Model is a discrete Spatio-Temporal Dynamic (STD) system based on local rules that can generate very complex STDs from relatively simple rule sets. It partitioned space into discrete units (*e.g.*, raster cells) and time into discrete steps. A rule sets specifies the state of each cell in time ($t+1$) based on the states of its neighbor in time (t). The system starts with an initial seed of cellular states and then iterates over discrete time steps, updating the state of each cell based on the previously specified programming transition rules. Therefore, CA can scale to address very large and detailed applications in many scientific fields rather than geography such as medicine, space physics, mathematics, and artificial life in computer science.

Based on Tobler's description of the theoretical foundation for a cellular geography using land use modeling dynamics as his applicable particular example, he assumes that the geographic space is represented as a lattice or raster tessellation (Tobler, 1979). Let (g_{ij}^{t}) be the growth of land use or land cover (LU/LC) at location (i, j) at time (t) - for example, classes such as: agricultural, urban, rural, and etc. -. There are five different types of models (Figure 3a) that can describe the change in the LU/LC at location

(i, j) from time (t) to time ($t + \Delta\, t$) as explained in the following equations (Eq$_s$.):

i. **The Independent Model:**

$g_{ij}^{(t+\Delta t)}$ (Eq.1); is a random variable that has no relationship with g_{ij}^{t}.

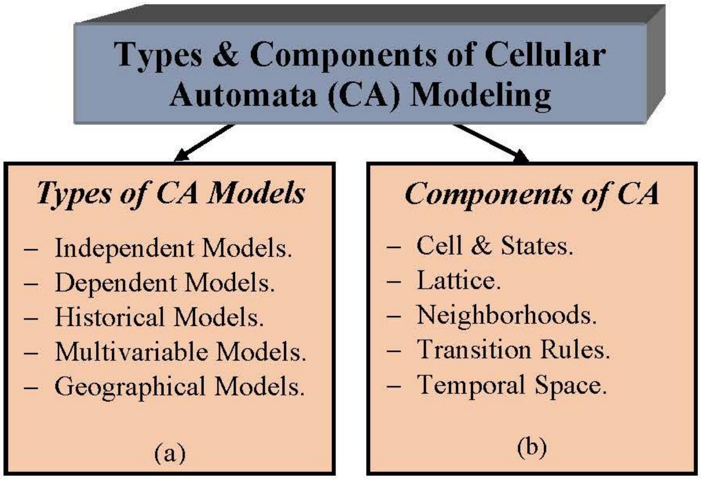

Figure 3. Types and Components of Grid-based Cellular Automata Modeling: (a) Types of CA Models. (b) Components of a CA model.

ii. **The Dependent Model:**

$g_{ij}^{(t+\Delta t)} = f\left(g_{ij}^{t}\right)$ (Eq.2); in other words, in this model the growth of LU/LC at time $(t + \Delta t)$ is a function of LU/LC at time t.

iii. **The Historical Model:**

$g_{ij}^{(t+\Delta t)} = f\left(g_{ij}^{t}, g_{ij}^{(t-\Delta t)}, g_{ij}^{(t-2\Delta t)}, \ldots \ldots . g_{ij}^{(t-k\Delta t)}\right)$ (Eq.3); in other words, in this model, the growth of LU/LC at time $(t + \Delta t)$ is a function of several previous LU/LCs at that location. This is also known as ***Time Series Model***.

iv. **The Multivariable Model:**

$g_{ij}^{(t+\Delta t)} = f\left(u_{ij}^{t}, v_{ij}^{t}, w_{ij}^{t} \ldots \ldots \ldots . z_{ij}^{t}\right)$ (Eq.4); in other words, in this model there are many other variables included such as (u, v, w, z, and etc.). Thus, the growth of LU/LC at time $(t + \Delta t)$ is dependent on several other variables at that location. This model can also be described as a system of simultaneous equations.

v. **The Geographical Model:**

$g_{ij}^{(t+\Delta t)} = f\left[g_{(i\mp p_i, j\mp p_j)}^{t}\right]$ (Eq.5); in other words, the growth of LU/LC at time $(t+\Delta t)$ is dependent on LU/LCs at other locations within a proximal area (defined by the parameter *p*). Moreover, the geographical model includes two possible sub-models. The first is an ***Extrapolation-Filtering Model*** - also known as ***Kriging model-***. It follows the main geographical model that described by the equation above. It is used to estimate and interpolate the missing unknown raster cells of LU/LC at location (*i,j*) depending on the LU/LC at other locations within an arbitrary user defined proximal area. The second sub-model is the ***Dynamical Geographical Model*** that is a special case of Kriging Model: $g_{ij}^{(t+\Delta t)} = f\left(g_{ij}^{t}, n_{ij}^{t}\right)$ (Eq.6), where n_{ij}^{t} is a short-hand for all LU/LCs within the ***neighborhood*** of location (*i,j*). Based on this concept, the cellular automata model can be viewed as a generalization of the dynamical geographical model.

2. *Cellular Automata Modeling Characteristics*

A simple explanation of the component and property characteristics of CA modeling, can be discussed based on many scientific efforts by many researchers in this field. Inspiring by (Batty, 2000) and (Schatten, 2018), the components and properties of CA can be discussed as follows:

2.1 *CA Modeling Components:*

A cellular automaton model consists of five basic components (back to Figure 3b) to be designed, constructed, and built it, as follows:

i. **The cell and states:**

The cell represents the smallest areal spatial unit shape of a cellular automata. As yet, the square shape is the most commonly and usually chosen in urban growth applications, but a triangular and hexagonal shape can be also arbitrarily chosen. The single cell is considered as the basic element of a CA Model. The cell size is a big concern in a raster-based application. Thus, it varies from one designed model to another. The size of a cell is considered as a "cell spatial resolution" that should represent the smallest meaningful entity in a geographical space in the earth's surface. The cell size is arbitrarily defined in a way to fit the observed geographical data. It can be in a size of smallest entity of a satellite image, which is in this case called a "pixel", or it can be for example in a size of an individual land parcel, buildings block, or a pre-defined area such as a residential land use lot.

Because the cell is a memory element, it can store different states. In its simplest case, there are two or binary states of a cell. Each cell can have the binary states that define its configuration that would be either "zero or one - 0 or 1-", dead or alive", "off or on", "inactive or active", "no-data or data". Thus, for unsophisticated settlement/urban growth model, a binary state would be "built up or non-built up"/"urban or non-urban". In more complex simulation, the cells can have more different complex states in a way that each cell can have more than one attribute and each of these attributes can have two or more states depending on the cell property. In the case of settlement/urban growth applications, cell can have various states, such as land covers, land uses, population, and degree of development.

ii. <u>The Lattice</u>:

A lattice is a raster space "grid tessellation" that consists of cells; or in other words it can be defined as the grid that contains of regular square cells. It is an arrangement of cells in a regular and discrete spatial configuration. The most common used lattice's dimensions are the one and two dimensional-lattice (1D & 2D) of CA Model, and the most sophisticated are the higher-dimensional automata. The 1D-CA is lined up in a string and has the privilege of being easy to visualize and has less possible set of rules. Whereas the states of one-time step are plotted in one dimension, the dynamic development can be shown in the second dimension. In a 1D-CA Model, the states can be shown in a flat plot illustrating the states from time step (*0*) to time step (*n*), while the 2D-CA model's visualization complexity and difficulty refers to its limited ability to show the state of only one time step in a two-dimensional plot. Therefore, most CA theoretical research papers were interested to deal with the properties of 1D rather than the 2D. Despite the modeling complexity of 2D-CA, most practical and applicable research papers usually functioned it to obtain many states. Therefore, the 2D-CA is frequently used in the urban growth modeling. Its lattice arranged as a Regular Square Grid (RSG) as the raster tessellation. It is applied to divide the space into uniform square cells. Although, these are more difficult to be visualized without some type of dynamic cartography or animation, their final results are remarkable, reliable and significant.

iii. **The neighborhoods**:

To transfer the arrangement of cells in a lattice from the static state to the dynamical state in the system, we have to add rules that describe how each cell's state should be updated. These rules are responsible for defining the state pattern of each cell for the next time step based on the states of the cells in its neighborhood. For the 2D-CA Model, there are five common types of local neighborhoods that were defined by Batty (Batty, 2000). These types are: Von Newmann, Displaced Von Newmann, Moore, Extended Moore, and Asymmetric (Figure 4); as follows: -

a. ***The Von Newmann neighborhood***:

A neighborhood with radius from the central cell is equal to one ($r = 1$). The central cell is also called "the target or focal" cell. The neighborhood cells consist of four adjacent cells taking the shape of a cross or following the main four geographical directions (N, S, E, and W); the cell above & below and to the right & left of the "target" cell.

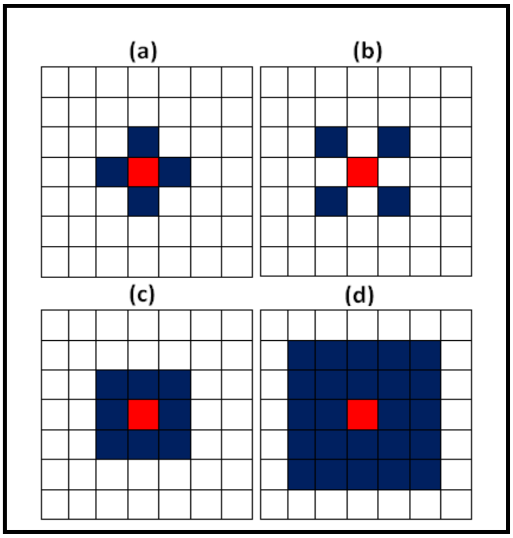

Figure 4. Local Neighborhood: (a) Von Newmann; (b) Displaced Von Newmann; (c) Moor; (d) Extended-Moor (source: Schatten, 2018; Batty, 2000).

b. *The Displaced-Von Newmann neighborhood*:

A neighborhood with radius is also equal to one ($r = 1$). The central cell is the target cell just as well. In a 3×3 cell array, the neighborhood cells consist of four corner cells. the pattern of this neighborhood is a symmetrically displaced version of the previously mentioned neighborhood type.

c. *The Moore neighborhood*:

This neighborhood has a radius is equal to one ($r = 1$), and its central cell is the target cell. The neighborhood cells in this type of neighborhood are considered as enlargement of the Von Neumann neighborhood to include the diagonal cells. In other words, the Moore neighborhood contain all 8 cells surrounding the "target" cell in an array of 3×3 cells (cellular space), which is the most usual and common neighborhood.

d. *The Extended-Moore neighborhood*:

It follows all the characteristics of Moore neighborhood except for the radius. Its radius from the central "target" cell is equal to two (r=2).

e. *The Asymmetric neighborhood*:

This is a user-defined, "arbitrary" neighborhood that is designed by the modeler. Yet, it should be guided by the four common local neighborhood cases around the target cell. It maintains the contiguousness of the cells in an odd numbered sizes of grid cellular space of 3×3 *(r = 1)*, or 5×5 *(r = 2)*. The Asymmetric neighborhood, should also conserve the cell contiguity within the cellular space. Figure 5, shows some examples of defining local arbitrary neighborhood cases based on some hypothetical user defined neighborhood to construct a CA model.

There is a positive linear correlation between " the number of the neighborhood cells including the target cell" and "the total number of possible binary state patterns of cell automata" that affects the transition rules. For example, in the von Newmann local neighborhood pattern that consists of five cells, the binary states (dead or alive) contain ($2^5 = 32$) possible cell patterns. For the Moore neighborhood

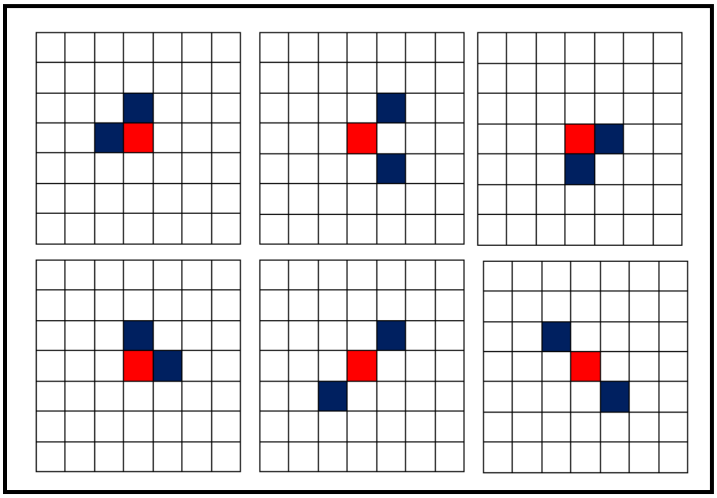

Figure 5. Local Neighborhood: Examples of different Asymmetric neighborhood.

with binary states, there are ($2^9 = 512$) possible cell patterns, in other words; there are 512 different configurations of on/off cell patterns that affect the transition rules. The neighborhood is usually constant across all cells and static. Yet, in a complex urban growth CA, the neighbor often indicates the relationship of land cover, land use, human activity, and other geographical conditions.

iv. The transition rules:

Some researchers considered the transition rules as the heart of CA modeling. A set of transition rules determine the state transitions of each cell as a function of the states of neighboring cells. Based on this definition, the transition rules determine the state of a cell in time (*t*+*1*)

is based on the pattern of cell states within its neighborhood at time *t*. In other words, it determines how the cell changes its state in the next time period based on the present state of the target cell itself and the states of its neighbor-cells just as well. Thus, each discrete sequence of time such as (*t* + *1*), (*t* + *2*), and (*t* + *n*) updates the states of the cells in question based on its transition rules.

Therefore, transitions rules represent the logic of the process which is being modeled, and thus determine the spatial dynamics which result. The set of transition rules are finite and constant across all cells. In 2000, Batty explained that the transition rules must be uniform and they must apply to every cell, state, and neighborhood at all times. Moreover, the change of state must be local, which in turn implies that there is "no action at a distance" (Batty, 2000).

The number of possible transition rules can be enormous. Mathematically, the number of possible rule sets can be calculated based on the resultant number of possible cell patterns by using two sequential equations. The first determines the value of (p), which is the number of possible cell pattern in the cellular space, as follows: $p = \Omega^h$, where (Ω) is the number of possible states, and (h) is the size of the neighborhood. Giving an example above of von Newmann neighborhood, the possible cell patterns is the value of "32". Given these patterns, there are different transition rules for the cell can be determined by the equation: $r = \Omega^p$, where (Ω) refers to the number of possible states, and (p) refers to the number of possible cell pattern that calculated from the first equation (Hurd, 1987).

By applying the same von Newmann's example, the possible rule sets are equal to ($2^{32} = 4{,}294{,}967{,}296$), which is a tremendous number. Stephan Wolfram determined the 256 elementary rules associated with a binary 1D-CA with a radius =1. The 256 "elementary" cellular automata have become to be known as "***Wolfram rules***", and the numbers associated with each rule are now often known as "***Wolfram numbers***". He showed that these elementary rules can generate patterns of enormous complexity (Wolfram, 1984).

v. **The temporal space:**

Cellular automata modeling process works through specifying some conditions that determining the start and end points of the simulation processing in space called "***initial conditions***" and time called ***"boundary conditions***" respectively. On one hand, the initial conditions are applied to the spatial configuration of cells and their states which "*start*" the process, as well as the time at which the process begins (Batty, 2000).

These conditions specify the starting point for the configuration of cells. While, on the other hand, the boundary conditions specify a stopping rule refer to limits on the space or "*time over*" or "*end*" the process which the CA Model is allowed to operate. The temporal space is the number of iteration processes that consists of both initial and boundary conditions to accomplish the simulation process.

Thus, it is considered as the time steps - start and end -in which the transition rules would be iterated and evolved. Notably, in temporal space, the temporal conditions are forced and dictated by the spatial conditions in that once the process begins from time zero, it ends and finishes when the predefined spatial boundary is reached.

2.2 *CA Modeling Properties:*

There are various number of physical and evolutionary properties concerning cellular automata modeling (Batty *et al*., 1999; Cochinos, 2000; Couclelis, 1985; Green, 1993; Torrens, 2003). Some important summarized properties are as follows:

1. Self-organization: Even when starting with a random arrangement of states, many (but not all) rules can generate complex pattern.
2. Life-like behavior: It follows the two basic fate states of the whole humanity of all time. The fate of any initial configuration of a cellular automaton is either to die out or to live. The living state has three different configurations. The first is to become stable or cycle with fixed period. The second is to grow indefinitely at a fixed speed, while the third is to grow and contract irregularly.
3. Self-Reproduction.
4. Develops in space and time.
5. A cellular automaton is a discrete simulation model; hence space and time are defined in discrete steps.
6. In one-dimensional automata, the cells lined up in a string.
7. Arranged in a two or higher dimensional lattice for two or higher dimensional automata.
8. The number of states of each cell is finite.
9. The states of each cell are discrete.
10. All cells are identical.
11. The future state of each cell depends only of the current state of the cell and the states of the cells in the neighborhood.
12. The development of each cell is defined by the transition rules.
13. Evolution of these transition rules leads to dynamical patterns.

The important implication of these properties is that models of complex systems need not to be complex themselves. Whereas, simple rules can generate the complex behavior that associated with the complexity of the cities to monitor many aspects such as urban growth.

2.3 *The Development's Mechanism of a CA:*

According to previously discussed CA characteristics of components and properties, the development of a pixel-based cellular automaton is controlled by a number of defined transition rules. These transition rules are formulated using simple Boolean statements and probabilistic expressions such as {< IF >, < THEN >, < ELSE >}. However, despite this simplicity, these simple rules can generate the most sophisticated and complex patterns of the cell development. Based on Wolfram rules, the CA as previously mentioned, is a discrete dynamic system in which space is divided into regular spatial units called grid cells. Because the state or value of the cells may change over time and time progresses in discrete steps, each cell in the system has one of a finite number of states. The state of each cell is updated according to local rules, *i.e*. the state of a cell at a given time depends on its own state and the states of its nearby neighbors at the previous time step (Wolfram, 1984). Therefore, the development of a cellular automaton is controlled by a number of transition rules. These transition rules are usually expressed as a set of "IF-THEN" statements. However, these simple rules can generate simple, moderate, complex, and even extremely complex patterns of CA development. Figure 6, clarifies an example of Moor neighborhood that a simple CA development pattern is generated with no transition rules limitations are required in a plain spatial area that has no geographical obstacles.

As a result, cellular automata are becoming very popular in geographic research for a number of reasons. One is that they are inherently spatial domain which is the main concept in geography.

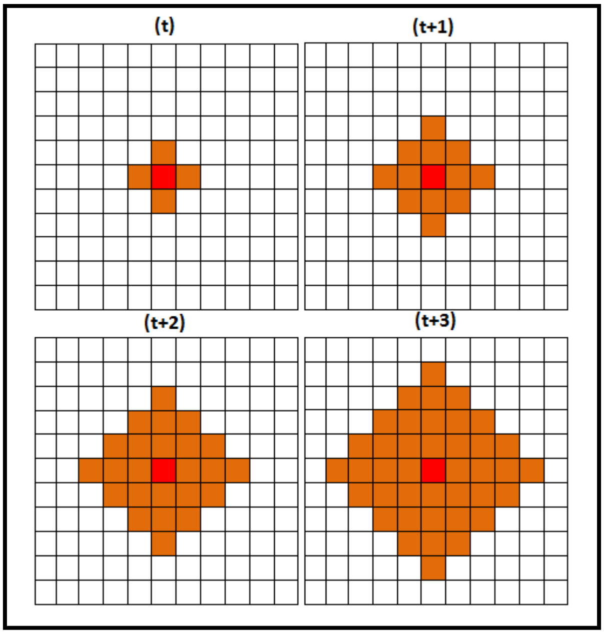

Figure 6. Generated CA for urban development that has no transition rules limitations following Moor neighborhood in a plain area with no obstacles.

A second reason is that they are computationally efficient and can be applied to problems with very high spatial resolution. There is also a natural link to GIS. The *Geographical Information Systems/Science (GIS$_{S/Sc}$)* provides a systematic and scientific platform for managing the spatial data required for CA. In return, CA allow GIS to go beyond static, geometric representations to include non-localized spatial process such as spatial organization, configuration, pattern, dynamics, transformation, and change (White & Engelen, 1997).

Comparably, conventional urban growth modeling approaches are mostly based on theoretical concepts that have many significant weaknesses rather than recent intelligent modeling techniques. The traditional models have a lack of considering various capabilities and factors such as: spatial-temporal dynamics capability, coarse representation of data, behavioral geographic factors, governmental policies factors, economic factors, and top-to-down approach, which ultimately fails to reproduce realistic simulations of urban systems. Some recent created CA models overcome some of these shortages, but not all together. Whereas, CA-based models are significantly handling the growth's dynamic spatial simulation and prediction that can integrated with Remote Sensing and GIS to deal with detailed or high-resolution modeling. Despite the complexity of these models, it can represent and display many complex real-world growth different patterns with simplicity. In this research, the SLEUTH urban growth model (SLEUTH-UGM) is chosen to be assessed and modified based on filling some basic shortages gaps.

VI. Conventional Vs. Modified SLEUTH Urban CA Models

Conventional theoretical-based urban growth models have a strict limitation in their abilities to generate meaningful future predictions for different rapid urban growth cases. These models failed to deal with the urban dynamics that characterize many cities and regions around the world. They ignore putting into consideration many "human-caused phenomenon" factors such as geographical, socio-economic, or even the frequent changeable central government policies factors that specifically occur in developing countries.

Thus, growth reasons differ from developing and developed countries. As for some developing countries that are suffering from the steadily increasing number of populations in their central cities, most rapid urbanization is still a process of a substantial migration from rural to urban cities. While on the other hand, many cities in developed countries are also suffering from the enormous number of populations in their cities and facing a huge explosion of growth through urban sprawl as a demand for more living space. The increasing urban-space consumption that is usually expand at the expense of fertile agricultural land essential for food production caused an increasing rate of urbanization as in the case of a country such as Egypt for example.

Moreover, the urban transitions and urban spread have evolved from being small towns toward being regular cities toward being metropolitan areas toward being megalopolises toward ending with the terminology of "Gigalopolises" as an indication of super cities which have been termed by "project Gigalopolis" that was also based on the work of Clarke and Gaydos in their published scientific paper in 1998. Therefore, urbanization and urban growth go hand-in-hand, and generate many other land transitions, with several varied land use types eventually converting to urban use (Clarke & Gaydos, 1998; Gigalopolis, 2003).

1. *Conventional SLEUTH Characteristics*

The SLEUTH cellular automaton urban growth model was created originally to predict regional patterns of growing urbanization. Its rules were more complex than those of other typical CA models with the involvement of using multiple data sources such as road networks, topography, and existing settlement distributions (Clarke, *et al.*, 1997). Although the designed functionality of SLEUTH was applied first on a specific geographical area such as "San Francisco Bay", it was intended to have a designed built-in growth rule that would be general enough to allow it to be applied also to other areas. Therefore, it was succeeded to be applied world widely in both developed and developing countries. However, it also failed to accomplish some applicable satisfying results from many other geographical regions. Thus, one major criticism of the model's first generation is its specificity to American cities and regions to which it was first applied.

It is well known that SLEUTH is designed to be a self-modifying CA model. The urban expansion in SLEUTH is usually modeled in a spatial two-dimensional grid and the basic growth procedure is a Cellular Automaton. Figure 7 illustrates the SLEUTH modeling controlling coefficients (the parameters setting that control the behavior of the system) and the types of urban growth to be simulated by the model. Six coefficients are controlled the behavior of the CA in the system. The main controlling parameters are: diffusion

(Dispersion), breed, spread, road-gravity, and two further controlling coefficients such as slope and user-defined parameters. Moreover, five types of growth (decision rules) are possible in the model, such as: spontaneous (Random), diffusive (new center), organic (edge), road influenced, and slope resistance type of growth (Gigalopolis, 2003).

Furthermore, SLEUTH is capable to model (Urban/Non-urban) dynamics as well as (Urban/LULC) dynamics. Both capabilities have

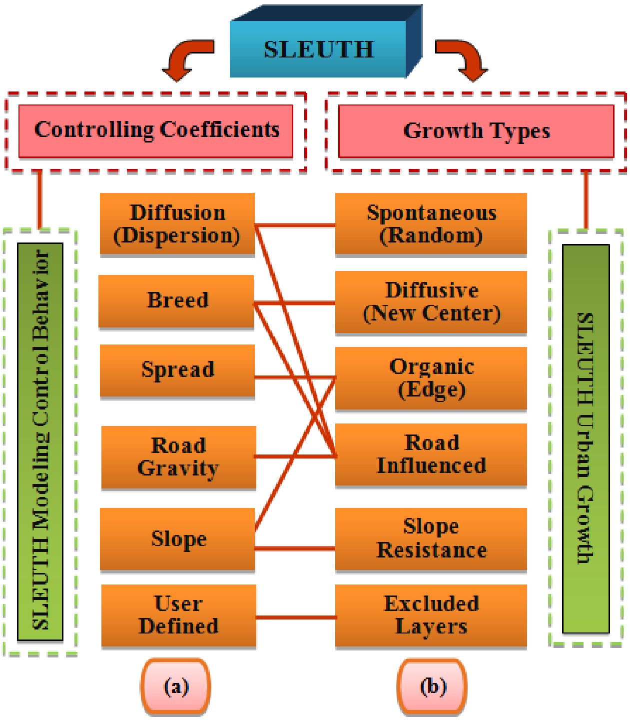

Figure 7. Simulated SLEUTH urban growth model: (a) Measuring growth control parameters (Controlling Coeff.). (b) Decision Rules of growth types.

led to develop two main subcomponents within the model (sub-models); an Urban Growth Model (UGM), and Land Use/Land Cover Change Model (LCD). It uses same calibration routine for each subcomponent model. Either one of the sub-models is analyzed one at a time. In one hand, if only urban growth is analyzed, then LCD is not activated by the model. On the other hand, LCD is activated when land use or land cover is being analyzed.

As was mentioned previously in the text, SLEUTH was originally created and written in C programming language that were ported for UNIX operating system platform. The general structure of SLEUTH model is illustrated in figure 8. It is originally created to have three main modeling stages. The first is *"the Initial Conditions"* that is developing the input dataset (coefficient values, cellular automaton seed, and input grid-based data). The second is *"Generate Growth Cycles"* that is concern with model calibration (setting parameters, applying growth rules, and self-modification). While, the third stage is *Concluding Final Simulation & prediction* followed by the final modeling output result.

Figure 9, displays on details the second stage of SLUETH model's general structure, which is the *Generate Growth Cycles* stage. It is starting with Data Input and ending with Final Modeling Output and it composes of two main components: the first is "Model Calibration & Parameters' Initialization", while the second is "Growth Rules & Model Prediction". The growth cycles stage was generated with sets of predefined growth rules that were applied in a set of nested loops. The outer control loop executes each growth history, keeping in mind cumulative statistical data. Meanwhile, the inner loop executes the growth rules (urban growth types) for a single year only. Thus, in this intelligent CA-based model, the urban areas behave like a living organism, trained by transition rules that influence the state of changes within the cellular automaton as a set of these nested loops. The growth rules are operated on a cell-by-cell basis, and the array is updated at the end of each year synchronously. Moreover, the calibration process uses a "***Brute Force***" calibration technique, which sequentially narrows down the range of SLEUTH behavior's parameters values, leaving the set which best replicate the historical data (Silva & Clarke, 2002; Yang & Lo, 2003; Chaudhuri & Clarke, 2013). Back to figure 7, SLEUTH simulates different growth types according to the growth rules. It encloses various growth cycle of different growth types with specific controlling coefficients. As for the spontaneous type of urban growth that randomly selects potential new growth cells is controlled by the diffusion coefficient, while the diffusive growth is controlled by the breed coefficient to allow the new spreading and disconnecting urban centers to let them start their own

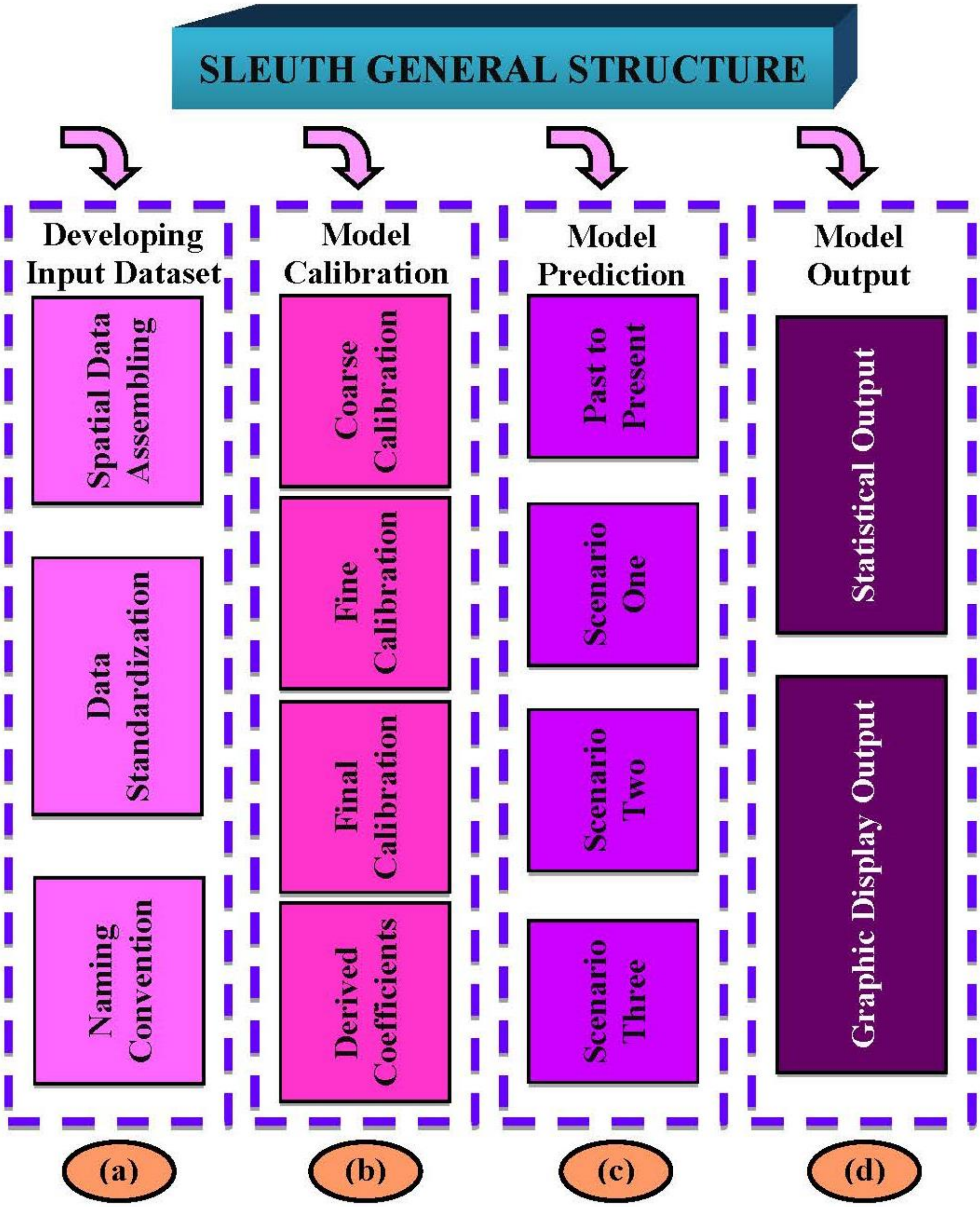

Figure 8. General Structure Procedures for the SLEUTH model Implementation: (a) Initial Conditions stage {preparing and developing different input datasets}, (b) General Growth Cycles stage {selected method for modeling calibration}, (c) & (d) Conclude Final Simulation and prediction stage {the final model resultant future prediction output according to user-defined time period}.

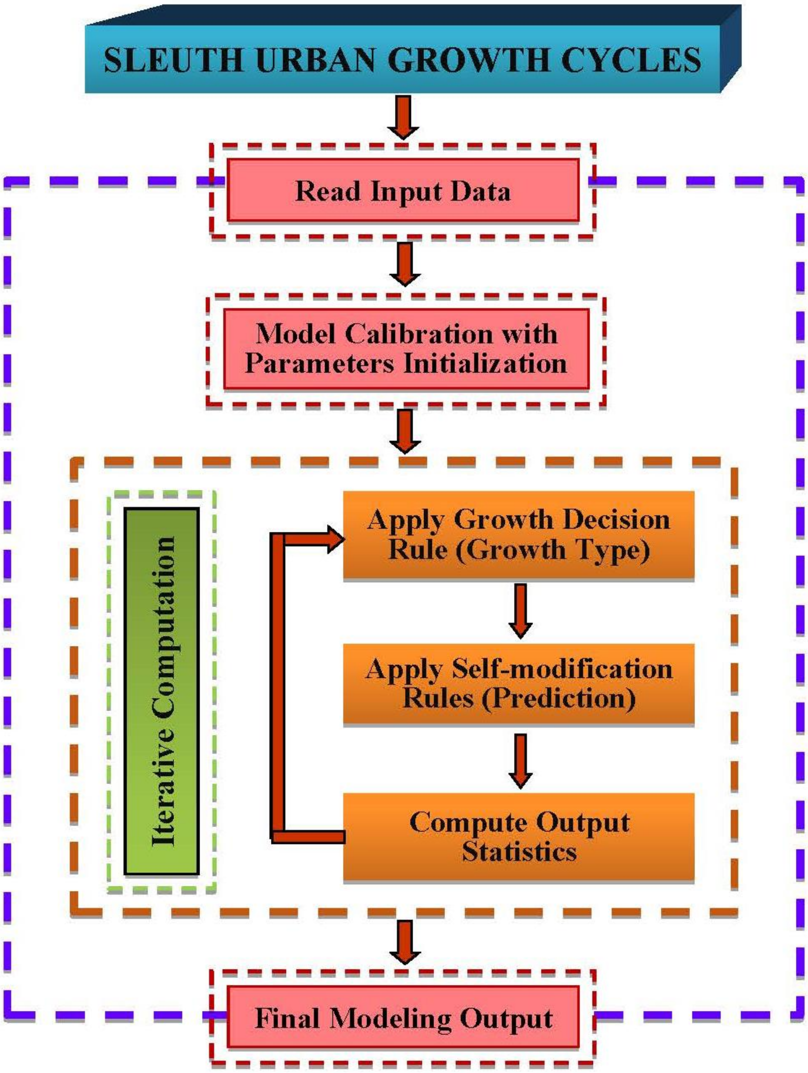

Figure 9. Generating "Urban Growth Cycles" stage of the SLEUTH model (adopted by Yang & Lo, 2003).

growth cycle. The edge growth type is controlled by spread coefficient to control the amount of outward "organic" expansion, while the road- influenced type of growth is controlled by three coefficients together (road-gravity, diffusion, and breed) to encourage the newly urbanized centers near or along the road network. Finally, the slope resistance growth type is controlled by the slope coefficient and its effects on reducing probability of urbanization prediction, while the excluded layer type is a user-defined coefficient that is conditioned by the user to specify the selected areas to be excluded from urban potential development (for example: deep water bodies, agricultural land, and strategic areas such as train stations and airports).

Thus, the growth rules evaluate the properties of the cell itself according to its neighbors. For example, if the neighboring cells are already urban or not, examining what their topographic slope status are, and examining how close they are to a road. In this way, simulation and prediction processes can be generated to fit each particular growth type to its simulated urban expansion according to the calibration of different self-modification growth scenarios of SLEUTH model.

2. ***The Operational Modifications Attempts of SLEUTH Model***

The notion of technically modifying SLEUTH is not a newly suggested idea by the geosciences' modelers and researchers. As being a programmatically open-source code stand-alone model, it attracted many researchers around the world to use and apply the model in their own study areas. On one hand, some researchers applied SLEUTH and have obtained significant results. On the other hand, many other researchers applied SLEUTH and published their scientific papers with no consideration to their fragile output results, and some others were stopped by their output misleading results and discussed this failure.

While testing SLEUTH world widely, some successful technical attempts have been practiced to enhance the model from different perspectives such as: efficiently reducing its computational time, enhancing its temporal sensitivity, improving the performance of the model and efficiently increasing the confidence of its results, and enhancing the applicability of the model according to the differentness geographic area that was applied (Chaudhuri & Clarke, 2013). Therefore, examined below are some selected technical and operational modifications of SLEUTH that were generated by Clarke himself and some other researchers to fill some technical gaps in the existed operated model and to overcome some modeling function, calibration, computational run time, and power shortages. Table 1. displays the main operational and conceptual modifications of SLEUTH Urban Growth Model (UGM) in the last 17 years. These modified Models are as follows:

2.1 *Optimal SLEUTH Metric (OSM)*:

In 2007, Dietzel and Clarke proposed a new optimal set of metrics to improve the calibration phase of the original SLEUTH modeling. They generated an exhaustive calibration parameter set for three synthetic test datasets to represent the major spatial processes and patterns of urban growth. For visual interpretation, they used a Self-Organizing Maps (SOMs) technique to explore the multi-dimensional parameter and metric spaces for the captured multiple datasets. Thus, these datasets were captured a wide array of spatial patterns. Their analyses and final results revealed a new metric for a modified SLEUTH model calibration that was named as OSM (Optimal SLEUTH Metric) measurement (Dietzel & Clarke, 2008).

2.2 *pSLEUTH*:

In 2010, Guan and Clarke developed a parallel version of SLEUTH. Their main objective is to improve the computational performance of SLEUTH, specifically during the calibration process. They developed and applied an open-source general-purpose parallel Raster Processing Programming Library (pRPL). This library is suitable to be utilized by GIScientists that have basic programming skills with less experience of

Table 1. The Operational & Conceptual Modifications of SLEUTH Urban Growth Model (UGM) in the period of (2007-2024).

Model Name	***Year***	***Developers***
Optimal SLEUTH Metric (OSM)	2007	Dietzel and Clarke
pSLEUTH	2010	Guan and Clarke
SLEUTH-3r	2010	Jantz *et al.*
SLEUTH-GA	2011	Clarke-Lauer & Clarke
PSO-SLEUTH	2019	Chandan, *et al.*
DSLEUTH	2019	Chaudhuri & Foley
SLEUTH-big data	2020	Clarke & Johnson
SLEUTH-CNNs	2024	The researcher

parallel computing and programming. The pRPL provides generic support for computationally intensive raster- processing algorithms that include local, neighborhood, regional, and global-scope algorithms as long as they are parallelized. Thus, it enhanced the capabilities of SLEUTH to work efficiently with massive raster data in a shorter period of time. Moreover, their modified method is proved to be qualified for replacing simplified assumptions during the calibration stage with a more exhaustive calibration process. This can produce different "*Best-Fit*" parameter combinations which help producing different simulation results. The results of this modification showed not only a great reduction of computation time for calibration process, but also worked with multiple processors and increased the modeling computational performance (Guan & Clarke, 2010).

2.3 *SLEUTH-3r*:

In 2010, Jantz *et al.*, developed an upgrading version of the original SLEUTH called SLEUTH-3r within the project Gigalopolis as a regional version of the model. It is an open-source software package that is easily downloaded. They presented a fine-scale (30-meter resolution) regional land cover modeling system that introduced new functionality and fit metrics that substantially increase the performance and applicability of the model. They developed new methods that expand the capability of the original SLEUTH to incorporate economic, cultural, and policy information.

Moreover, the new modified modeling version allowed opening up new avenues for other Land Change Models to be integrated with it. Their main objectives are: (1) to improve the model's bias toward edge growth that limits and restricts the suitable level of diffused growth in fine resolution data; (2) to fix the inappropriate fit statistics; (3) to fix the inefficient memory use for larger datasets in SLEUTH. These objectives were addressed through direct modifications of the source code itself of the original modeling version (written in the C programming language), which allows the interactivity in setting the model coefficients.

Additionally, there is another objective that the researchers aimed to achieve, which is fixing the inability to identify areas where growth is more likely to occur in the limited original version. For this fourth objective, they developed a method to divide the region of interest according to groups of minimum variability of the urban growth pattern by using cluster analysis and simulating each pattern independently (Jantz, *et al*., 2010)

2.4 *SLEUTH-GA*:

The calibration mode or stage that is responsible of running SLEUTH model is considered as a corner stone of executing the process of modeling the change of urban form over time. Thus, most modeling modification attempts concern with testing and examining the best calibration methods to be conducted to give a better simulation and prediction result can be possibly achieved. In 2004, Noah Goldstein was the first to propose using "Genetic Algorithms (GA)" as an efficient calibration approach rather than using the original Brute Force Method (BFM) in SLEUTH. He proved in his research paper the superiority of GA calibration method to improve the model fits, as well as fulfilling computational needs (Goldstein, 2004). Four years later, a significant study was conducted by Shan and his researcher colleagues to suggest using GA to enhance the efficiency of transition rule calibration in SLEUTH. Their final results proved that GA will significantly benefit urban modeling problems that have larger set of input spatial data and bigger solution spaces (Shan, *et al*., 2008).

In 2011, Clarke-Lauer & Keith Clarke developed revolutionary Genetic Algorithms (GA) heuristic method based on the improvement of the mentioned above Goldstein's proposed method. The "SLEUTH" modified model still has three modes (test, calibration, and prediction). These modeling modes are used to implement the model in the selected applicable area of study. Basically, the "test" mode verifies the suitability of the input datasets before performing the calibration or the prediction modes. As for the "calibration" mode, it aims to find the best possible combination of growth coefficients. Whereas, the final modeling stage which is the "prediction" mode is using these calibration findings to simulate the urban growth into the future. In SLEUTH-GA modified model, the modelers redesigned and significantly improved the calibration of the model with adding choices on encoding, fitness evaluation and survival selection. They tested and examined their redesigned model and their final results showed that using GA for calibration has minor improvements in the goodness of fit of the model, but it also showed greatly decreases the computation time of calibration; by a factor of 5. Therefore, developing and testing SLEUTH-GA showed and proved the advantages of genetic algorithm calibration that include fully automating the calibration process and remove any remaining human subjective choice (Clarke-Lauer & Clarke, 2011; Clarke, 2017; Clarke & Johnson, 2020).

2.5 *PSO-SLEUTH*:

In 2019, Chandan with other researchers aimed to enhance optimization of transition rules based on machine learning techniques and evolutionary algorithms that follow nature-inspired mechanism. They integrated Particle Swarm Optimization (PSO) as a post calibration technique with SLEUTH. Their demonstrated results of the proposed modeling modifications were evaluated and validated using statistical fit measures. In a comparative analysis, the model performance was tested and compared statistically between the two (original BFM) and (suggested PSO) calibration methods. Their suggested calibration technique revealed a better performance comparing with traditional one. In addition, they achieved another significant development concerning decreasing computation time of optimized values to hours instead of days. Thus, the testing results show superiority of the suggested over the traditional one. Therefore, they modified SLEUTH codes with respect to PSO and they revealed their proposed modification *PSO-SLEUTH* model. They stated that their results are confirming the model robustness, as well as delivering significant fit statistics without compromising optimum coefficients (Chandan, *et al*., 2019).

2.6 *DSLEUTH*:

In 2019, Chaudhuri & Foley developed a distributed framework for SLEUTH and they named it as DSLEUTH. They modified the original dynamic model for executing and managing subsets of the parameter space on a shared memory machine, while the calibration time was decreased. Moreover, it allowed modeling larger datasets within a reasonable amount of time without compromising the accuracy of the simulation output. Their suggested modification describes the framework and a performance study using two different datasets; the first is data with grid size of (200×200) showed a 21x speed up, while the second is data with grid size of (906×1005) showed a 19x speed up. Their framework yielded a 20x speed up on both datasets using 40 processes (Chaudhuri & Foley, 2019).

2.7 *SLEUTH-big data*:

In 2020, Clarke & Johnson investigated the spatial consistency of SLEUTH urban growth and land use (UG/LU) change by using massive datasets to project the American state of California's land use for the entire 21st century (up to the year of 2100). Inspired by the work of (Zhou & *et al*., 2019), they developed the model by using partitioned data (tiles) to reduce the computational load in the calibration and enabled the model to allocate sufficient execution memory. They used high spatial resolution raster datasets (30m) and they divide the whole California state into 8 zones, four of them with 16 subtiles and the other four with 32. Eventually, 174 subtiles are used for calibration and forecasting after excluding the No-Data area, area fell in the ocean, and area that located outside the state. Calibration consisted of invoking the SLEUTH-GA code using the Cygwin UNIX emulator for MS Windows. In their research paper, spatial autocorrelation was found to propagate forward into the SLEUTH forecasts, resulting in major differences within the California state in the change rate of (UG/LU). The resultant best-fit coefficients from the calibration are displayed for each controlling coefficient (Spread, Breed, Slope, and Road gravity), and also the final modeling forecasting results are displayed and mapped by subtiles (Clarke & Johnson, 2020).

Furthermore, all modified UGM SLEUTH models still require a minimum of four historic data that contains urban extents outputs along with the other required data such as elevation, slope, exclusion layers. As for the transportation network data, the minimum requirement are two layers; one representing initial and final stages. all data input must have same geographic registration of Cartesian Coordinates, projection, and datum along with same spatial resolution that is required by the model to be (30m) for all data input.

In this research, the suggested conceptual modification of Keith Clarke's "SLEUTH" urban growth model (UGM) which is named by the researcher as: "SLEUTH-CNNs", is not only a proposal study that offers a new development of the current executive model, but it is also a suggestion to make a new perspective of the model to be evolved from being one easy-access open source model to be totally transformed to be a complete revolutionary open source code (intelligent modeling/multi-system) independent software package. In other words, it is a suggestion to draw a conceptual framework of a proposed innovative intelligent modeling as a free software which could be -in one hand- operated and worked individually as a complete stand-alone system. In addition, and on the other hand, it can also may integrate with another stand-alone intelligent system. Therefore, the researcher proposed that it would be easily to be integrated to an another newly suggested geospatial intelligent computer system that is named also by the researcher as: ***Geo-Information Intelligent Systems (GIIS)*** to simulate not only this examined urban growth modeling but also it can deal with any other complex geospatial models. About the suggested new intelligent system, the researcher will explain later in a further "research study" the concept and functionality of her proposed new intelligent system which is termed and abbreviated as **GIIS**.

Continuing the latest two modification attempts of the studied Clarke's model (SLEUTH-*GA* and SLEUTH-*big data*), the proposed modified urban growth model is named by the researcher after infusing the Deep Learning (DL) method of the ***"Convolutional Neural Networks CNNs"*** to the current SLEUTH model.

Conceptually, discussed below the proposed ***SLEUTH-CNNs*** new modeling modification and draw a complete modeling cycle that starts with inputting datasets, training method, classification, performing accuracy assessments of classification analysis, then proceeding the modeling algorithms and implementation in which the model is calibrated, simulated, and visually displaying the final predicted urban growth resultant outputs. All the suggested intelligent modeling algorithms is proposed to be all integrated and programmatically connected together with a "Code Plock" which is a C programming language's compiler to maintain the source and target language codes and to produce an only one stand-alone modeling multi-intelligent system software package. Moreover, after achieving the final prediction results, this research also offers a suggestion of some statistical measuring methods that would be used for modeling accuracy assessment. A schematic flowchart showing the proposed conceptual modeling modifications is finally illustrated.

VII. Proposed (SLEUTH-CNNs) Urban Growth Model

In this research, the proposed Cellular Automata based Urban Growth Modeling (SLEUTH-CNNs) is recommended to be still an open source for a free modeling usage for modelers with no financial profit, but also in the same time; instead of being an operational dynamic model, it should be transformed to be a non-profitable "stand-alone intelligent software package" as suggested and mentioned above. This complete operational multi-system software should consist of a variety of GeoAI algorithmic methods. These algorithms are suggested to be divided in two main sub-systems as follows: the first sub-system is the **"*data preparation & processing mode*"** of the suggested infused Convolutional Neural Networks (CNNs) with its calibration and validation methods, while the second sub-system is the ***"calibration & prediction modes"*** that consists of the predictive Cellular Automata (CA) modeling that is coupled with another intelligent calibration approach such as Genetic Algorithms (GA) that is coupled with CNNs. Explained earlier on details the characteristics of CA dynamic model with its three pillars of components, properties, and mechanism. The other modeling algorithms including the suggested infused one with its built-in internal algorithms are discussed as follows:

1. *Data preparation & Processing Modes*

1.1 *CNNs-based Deep Learning Intelligent Algorithm:*

A biologically inspired modeling technique, comes recently -in the past few years in Geoscience fields- the Convolutional Neural Networks (CNNs and sometimes ConvNets) based Deep Learning (DL) intelligent approach to become an attractive area of interest for many researchers in many scientific disciplines. The beginning was in 1989 at Bell Labs in New Jersey, when "Yann LeCun" developed with his colleagues the first operational Convolutional Neural Network type that was trained with back propagation (BP) and also was implemented with supervised training; the "LeNet", that could effectively recognize between different handwriting patterns (LeCun, *et al*., 1989). In 2006, Geoffrey E. Hinton and his colleagues explained the concept of artificial intelligence Deep Learning (DL) approach and they suggested that Multiple Hidden Layer Networks (MHLN) have better feature learning ability. They concluded in their research paper that the complexity in training process of MHLN can be effectively decreased through its layer's initialization (Hinton, *et al*., 2006). From this point, the Deep Neural Networks approach started to have the attention of many other researchers in many scientific fields.

In Geoscience and Geospatial Artificial Intelligence fields, Penatti with his colleagues were first to adopt "ConvNets" -as they named the CNNs in their research study- for both fields of aerial photography and remote sensing in various tasks especially the classification one. They practiced a preliminary set of experiments which indicated and concluded that the pre-trained ConvNets can be transferred to the remotely sensed image classification task and achieved significant and inspiring final results (Penatti, *et al*., 2015). Afterwards, CNNs has achieved a great success in remote sensing practical applications such as: visual recognition, image classification analysis (agricultural crop types, land cover and land use, vegetation extraction, built-up area extraction, and hyperspectral image classification), image segmentation, road tracking, multi-temporal time series analysis, object detection, and change detection analyses.

Basically, Deep Learning models are considered as a separate class of the Machine Learning (ML) algorithms that can intelligently learn a hierarchy of features by constructing and automating high-level features from low-level ones. Additionally, this learning machine algorithms can be trained using either supervised or unsupervised approaches, and the resultant systems have been proven to produce competitive performance in visual object recognition and audio classification (LeCun, *et al*., 1998; Lee, *et al*., 2009). In other words, Deep Learning approach allows the computer system to build many representations out of simpler ones. Generally, it is known as "representation learning". In addition, it has solved large scale increasingly complicated applications with increasing high accuracy. Thus, the deep learning based CNNs intelligent method is boomed in the last five years -before the publishing date of this current research paper- for image recognition, classification, pattern recognition, and modeling future settlement growth especially the urban growth one.

1.2 ***CNNs Training Algorithms:***

As for the training methods of classification analysis, it is divided into two main types, one is the supervised learning training and the other is the unsupervised one. The modeling architectures of the supervised learning algorithm are the ***"Multilayer Perceptrons (MLP)"*** which is preferable to obtain the best desired classification resultant outputs with higher accuracy. The MLP is an artificial neural network that consists of a minimum of three fully connected layers of neurons (nodes); the input layer, the hidden layers, and the final output layer. Noticeably, the network processes the data inputs and compare its resultant outputs against the obtained pre-selected training sample sites for each class. Errors are then propagated back through the system, causing the system to adjust the weights which control the network. This process is repeated frequently as the weights (the selected parameters and thresholds) are continually twisted and reached its ideal statistical results **"best-fit"** and refinements. While the unsupervised learning algorithm provides the network with inputs with no selected training samples which make the system itself self-organized the data and must then decide what features it will use to group and cluster the input data to produce a number of output classification classes. The main architecture of the unsupervised learning is called ***"Kohonen network"*** which is useful in self-organizing clustering untrained data.

Hence, the supervised Deep Convolutional Neural Networks (CNNs) modeling algorithm is a type of an artificial intelligence deep learning models in which it significantly gained recently -especially in the past five years as mention above- a serious attention in the remote sensing science community. One main capability of CNNs is the powerful ability to automatically discover relevant contextual features in image categorization problems. This ability comes from its mechanism of having a group of intelligently learned convolution filters that help extracting these hierarchical contextual image features. Thus, with the increasing development of high-resolution satellite images, the needs for qualified intelligent remotely sensed classification analyses are getting more essential. The complexity of the newly complicated produced high resolution satellite datasets, give birth to a series of recent studies on the deep learning approach and on CNNs specifically. Figures 10 & 11 displays the intelligent modeling architecture and the functional structure of the Convolutional Neural Networks (CNNs) of both classification processing analysis and feature extraction process, respectively. Noticeably, in the field of pattern classification of GeoAI science, Deep Learning CNNs become one enlighten hotspot nowadays.

Therefore, the intelligent CNNs is adapted by the researcher herself to achieve the first stage of the proposed "SLEUTH-CNNs" model for classifying and monitoring LC/LU (Land Cover/Land Use) from the input grid-based datasets (multi-source/multi-temporal satellite images and other raster-based data). In addition, feature extraction process of

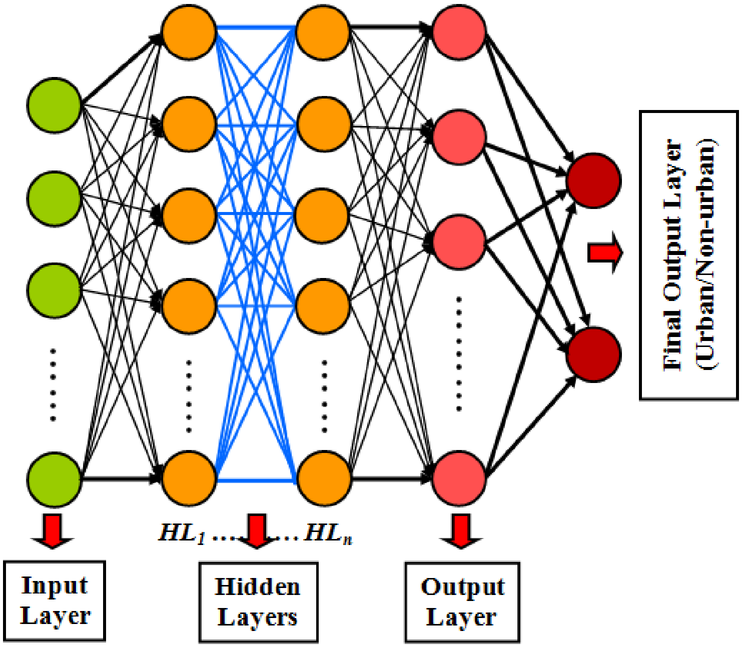

Figure 10. General Architecture of Supervised Deep Learning Multi-Layer Perceptron (MLP) based Convolutional Neural Networks (CNNs) for Classification Process Analysis.

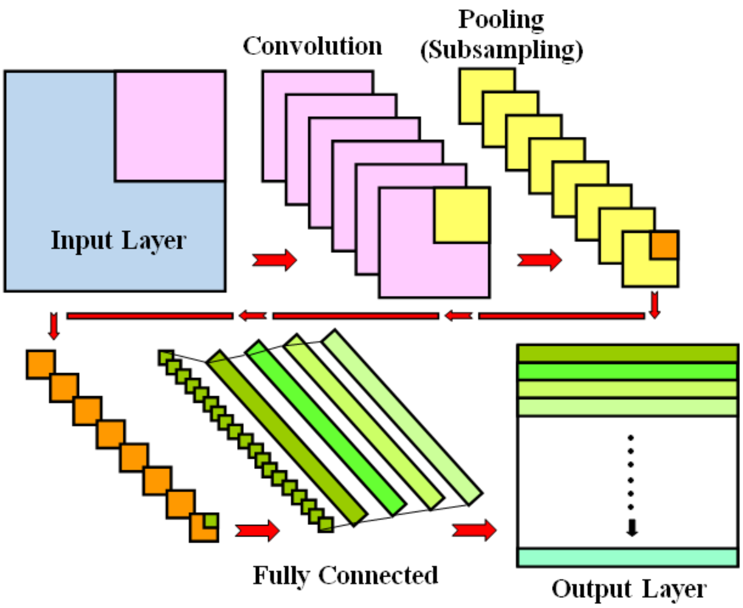

Figure 11. Structure of Convolution Neural Networks (CNNs) of Feature Extraction Process.

the urban/non-urban land cover is achieved to conduct the proposed modification of Urban Growth Model of SLEUTH. Due to the impressive capability of image features recognition that is managed by CNNs, it can accurately identify and classify satellite images or else with the highest accuracy assessment comparing with other intelligent methods. Additionally, one major advantage of choosing CNNs algorithm for this task, is to avoid all the complicated traditional pre- processing of satellite images which allows original source of data imagery (raw data) to be inputted directly into the intelligent algorithm and enhanced it with self-organization processes and self-accuracy assessments.

1.3 *MLP-based CNNs Functional Structure:*

The main functional structure of the CNNs is based on Multi-Layer Perceptron (MLP) supervised deep learning method and consists of three main parts of the intelligent algorithm layers and they are: ***{Input, Hidden(s), and Output(s)}***. In addition, there are four main algorithmic operations and functions and they are: ***{Convolution operation, Pooling operation, Activation function and Loss function}***. As for the different categories of layers in the Supervised Deep Learning MLP-based Convolutional Neural Networks (CNNs), they are infused with the convolution and pooling operations while processing. The input layers' part is mainly consisting of satellite images with moderate, high, higher, or even hyper (spatial, spectral, and radiometric) resolutions. However, it can also consist of different grid-based datasets with different raster or image data formats.

Whereas, the CNNs algorithm consists of a sequence of convolutional and pooling (subsampling) operations that follows each other. When these operations are computationally ended, it is followed by fully connected output layer(s). The convolution layers' extract features from the input one by computing the focused cell (pixel) that is orderly selected by the system to produce a new cell value which is resulted from the computation process between the weights and the system's specified kernel window of input data that would be then connected afterwards. The kernel windows' mission is to extract features and capture different patterns such as edges, textures, and much more complex image various structures. The resultant values are then passed on to the next layer for additional processing (Figure 12).

Sequentially, pooling operations produce pooling layers that are followed directly the convolution layers. Pooling layers is performing a down-size sampling (sub-sampling) operation along the width that end with a resultant dimensional reduction output layer. There are many reasons behind operating the pooling layers in CNNs. The main reasons are: reducing the number of parameters and decreasing computational complexities just as well, reducing the spatial dimensions along with decreasing the training time (number of computing time iterations), and addressing over-fitting concerns.

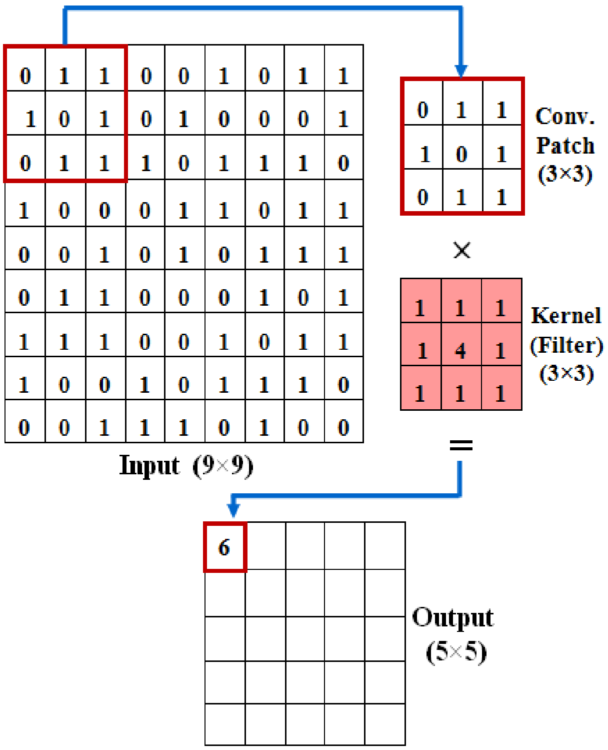

Figure 12. Resultant output (5×5) of Convolution Operation of CNNs between a 9×9 input and a filter of 3×3 kernel size.

There are many types of pooling operations according to the selected statistical methods that built-in the algorithm's source code. For example, Max-Pooling, Min-Pooling, and Average or Avg-Pooling (Figure 13). The Max-Pooling method is the widely practiced technique in the algorithm and it is operated by selecting the maximum cell value from the subset pooling window to assemble the pooling layers in later stage.

1.4 ***CNNs Calibration & Accuracy Assessment Algorithms:***

Regarding the calibration processing mode and assessing accuracies, the CNNs consists of two main functions as mentioned above. The first is the calibration non-linear algorithm methods that is titled as "**Activation Function**", while the second is concerning with measuring and assessing the modeling accuracy which is titled as

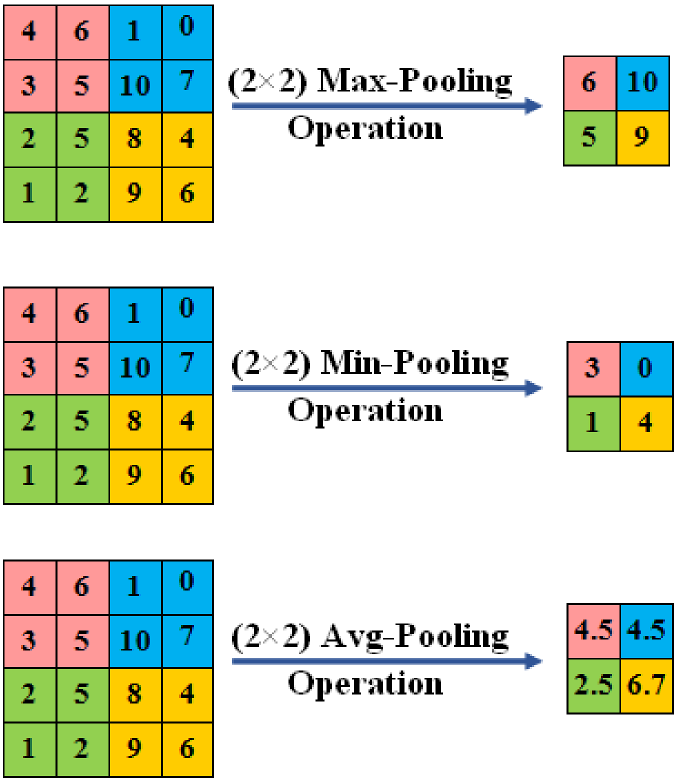

Figure 13. Examples of Different (2×2) Pooling Operations of CNNs.

"Loss Function". As for the first function, it is not only considered as a crucial foundation part in the CNNs design, but it is also one essential components of the Deep Neural Networks. In the non-linear function, each neuron in the convolutional neural networks applies an activation function to its net input to determine its output signal (Yadav, *et al*., 2015). Thus, it allows the algorithm to firstly recognize then secondly seized and captured the nonlinearity relationship that is associated between the input and the output layers. Therefore, without the activation function, it is noticed that the neural system would be forced then to function the linear transformation instead, which it will affect severely the system ability to model the complexities of data patterns recognition. Basically, there are some basic and widely used non-linear activation function methods that controls both hidden layers and output layers. The most common used methods are: the classical "Sigmoid" method, "Sigmoid Linear Unit (SiLU)", "Hyperbolic Tangent Function (tanh)", and "Rectified Linear Unit (ReLU)" activation function methods (Vakalopoulou, *et al*., 2023).

The second main function component of CNNs is the **"Loss Functions"** methods for measuring the modeling accuracy. This type of function is representing the error for a given prediction. It measures the difference between the predicted modeling output and the real ground truth data. In other words, the loss function specifies the classification errors by quantifies the difference between the predicted classified output and the actual target values. One major advantage of loss functions in deep learning CNNs is to minimize this difference. It directly effects the training process and the performance of the CNNs. LeCun specifies that one of the simplest output Loss Function that can be used with the CNNs is the "Maximum Likelihood Estimation (MLE)" method and he argues that this method is equivalent to the "Minimum Mean Square Error (MSE). However, in Deep Neural Network algorithms, the loss function is crucial to measure errors and delineate accuracies between both resultant output and ground truth data (LeCun, *et al*., 1998; Vakalopoulou, *et al*., 2023). Therefore, some other loss functions methods can be also effective, such as: Cross-Entropy (CE) Loss, Binary Cross Entropy (BCE) Loss, Complete Intersection over Union (CIoU) Loss, and the Distribution Focal Loss (DFL).

The researcher proposed certain operational layers and functions to be utilized in the CNNs algorithm that would be infused later in the suggested SLEUTH-CNNs proposed intelligent modeling multi-system. As for the input layers, it is preferably to unify the spatial, spectral, and radiometric resolutions of all inputted datasets (satellite images and other grid-based data) as an initial and a prior operation step that should be applied before functioning the algorithm itself. In one hand, this pre-processing operation is essential to specify the layers' configuration by determining the number of neurons based on the number of inputs in which the network will be handled in the CNNs algorithm. On the other hand, it is also crucial to be used in a further modeling step for a time series operation in SLEUTH-CNNs prediction mode.

Moreover, it is also suggested by the researcher to utilize a 2D convolutional filter with a 2×2 Max-Pooling operator. Furthermore, it is suggested to utilize also the "Sigmoid-weighted Linear Units (Si-*w*LU)" as an activation function rather than using any other method (Eq.7).

$$f(x) = W \times \frac{X}{1+e^{-x}} \qquad (EQ.7)$$

Where: $f(x)$ is the sigmoid function, W is the computed convolutional weights, X is the input to the sigmoid function, and e is Euler's number which is a mathematical constant that is equal to the number of 2.781. The "Si-*w*LU" equation is specifically improving the performance during training. Moreover, it is a non-linear activation function that is suitable for binary classification applications such as urban/non-urban classes. In addition, this non-linearity function allows the CNNs to model complex relationships between the modeling inputs and outputs. Another crucial advantage of using this "Sigmoid-weighted Linear Units" activation function method, is the capability of being differentiated at the value of zero which results in a smoother activation function (Elfwing, *et al*., 2017). As explained above, the fully connected layers are responsible of connecting neurons that attached in both previous and subsequent layers to create final predictions or classifications. In our case, we interested in the classification mode, therefore a final classification output layer should be resulted.

Finally, the Loss Function should be accomplished to measure the accuracy by reducing errors as much as possible. Thus, a "Distribution Focal Loss (DFL)" method is proposed by the researcher to assess the final accuracy of the CNNs resultant classification output. It is a mathematical function of the parameters of Deep Learning Algorithms (DLA) One major advantage of DFL is significantly handling class imbalance by reducing weights and distributing the focal loss across multiple classes and scales. This type of loss distribution allows balancing the learning process and making it more effective in complex detection scenarios that helps CNNs to detect easily the most challenging (difficult-to-classify) objects such as small or overlapping ones. In 2020, Li and his colleagues proposed a novel DFL method and its formulation that offers a distributional imbalance between correct and incorrect classifications consists of two kinds of distribution biases, which enforce the neural network to improve the object detection accuracy (Eq.8).

$$DFL(\text{û}, \text{u}) = \left[1 - \frac{|N\ (\text{u})|}{|U|}\right]^{y} \ \|\text{û} - \text{u}\,\|^{2} \qquad (EQ.8)$$

Where: û is the estimated uncertainty score, u is the ground truth, N (u) is a small continual subset consisting of samples with similar uncertainty, and the relative volume (density factor) of this subset: $|N\ (\text{u})|/|U|$ is a rational approximation of local frequency in the uncertainty distribution (Li, *et al*., 2021).

2. *Calibration & Prediction Modes*

The final sub-system of the proposed modeling approach is the modeling implementation that consists of both calibration and prediction modes. This suggested multi-system urban growth modeling is based on cellular automata (CA) to model, simulate, visualize, and predict the future behavior of this man-made phenomenon. The proposed sub-system keeps maintaining and preserving the basic modeling designed skeleton of Clarke's SLEUTH-GA urban growth model. In this stage, the suggested innovative modification achievement would be concentrated on modifying only the calibration method of the current model. Therefore, the SLEUTH's modeling implementation still requires two main phases. The first phase is the calibration to determine the best conventional SLEUTH's pre-defined controlled coefficients (diffusion, breed, spread, road gravity, and slope) that are based on the past growth pattern. These coefficients must reach its best-fit with obtaining the highest Optimal SLEUTH Metric (OSM) with a measuring values ranging from zero to 100 for each coefficient individually. Where the measuring value of "zero" means an absence of a behavior, while the measuring value of "100" means an unrestricted behavior (Clarke, 2018). The second and final phase is the predicting one to model the urban growth into the near or far future according to modeler-defined of predicting time period.

2.1 *Calibration of Cellular Automata Predictive Model:*

Basically, calibration mode that is built in any dynamic model is the most important learning stage which is always based on the past and present information to successfully predict future simulations. Thus, it forms the spinal cord of any dynamic predictive intelligent model. Urban growth predictive modeling algorithm needs to have time series temporal datasets to represent the historical characteristics of the urban growth, extract all past and present growth patterns with its full extent in each time period, and to measure its growth rate. By obtaining the growth rate history of the area of study, one can run the calibration mode in three different scenarios to estimate the various modeling coefficients. The three major growth rate scenarios are: Decrease, Increase, and Constant (Stable) -that was adopted by the traditional SLEUTH - growth rate. In SLEUTH-3r, the diffusion coefficient multiplier is changed from being only constant, to permit the user to change this multiplier value interactively; when it is increased or decreased, the number of urbanizations attempts for diffusion growth changes accordingly (Jantz, *et al*., 2010). From this perspective, SLEUTH was operationally modified and improved its calibration methods for better performance. The scientific comparison between performing traditional Brute Force (BF) calibration against using Genetic Algorithm (GA) was in favor of the GA calibration method. Clarke debated that both methods performed as well as each other, but the GA used vastly less computation time comparing with BF with speed up about 3 to 22 (Clarke, 2017). With the new and fast advances of rising geographical/geospatial artificial intelligence methods (GeoAI) that are infused lately in some complex and sophisticated dynamic geographical models such as CA-based urban growth model, the needs for integrating these models together for the purpose of improving its performance capabilities became an essential necessity nowadays. Therefore, the researcher suggests to couple the GA calibration method of the "calibration and prediction" modes with its equalized mode part of the CNNs proposed method.

2.2 *Coupling Genetic Algorithm (GA) & Deep Learning CNNs*:

The supervised GA is an evolutionary algorithm which is defined mainly as being an adaptive heuristic search algorithm that is usually used to solve optimization problems in Machine Learning. The successful initial attempt of switching Brute Force (BF) calibration method with the machine learning GA that was achieved by Clarke-Lauer and Clarke in 2011 to optimize SLEUTH performance and to minimize the modeler's interaction and self-judgment, has led later to enlarge the role of artificial intelligent (AI) methods in Geoscience modeling algorithms specifically in calibration mode. The derived

"SLEUTH-GA" modified model was tested later for more general applications and examined by setting more extra procedures, constant parameters, and programming codes to reach its final and complete new operational version by the year of 2018 (Clarke, 2017; 2018).

Enhancing the efficiency of transition rule calibration in Cellular Automata based Urban Growth Modeling is considered a vital scientific functional and operational concern that faced all modelers. In one hand, solving the modeling dilemma of finding the best combination of transition rule values that can match the real world's urban behavior and pattern is one of the most critical steps to validate the performance of this designed CA model. On the other hand, reducing the CPU computational processing time is also an essential calibration goal to be reached by the modelers. Therefore, moving toward replacing the old traditional "time consuming" statistical calibration methods with artificial intelligence ones become a dynamic modeling necessity nowadays.

Artificial intelligence Genetic Algorithms method was selected by Clarke-Lauer and Clarke as mentioned above to evolve and improve the SLEUTH's calibration algorithm. They succeeded to give a good example of experimenting GeoComputation technology, where computer science optimization methods such as GA meet simulation modeling in geography. They applied the GA at code level by replacing the source code that implements the old traditional statistical one along with maintaining and preserving all of the other model behavior modules. Their GA calibration process populates a (that is called a Chromosome) with a set of parameter combinations (that are called Genes), of each one of the required "previously mentioned" five control parameters that are already specified earlier by the model (Clarke-Lauer & Clarke, 2011; Clarke, 2018). There are three main types of genetic operators, and they are: Selection, Crossover, and Mutation. These operators must in conjunction with one another to develop a successful algorithm. Thus, GA is considered as a global search algorithm that is conducted through its operators to reach its generation during iterations, then evaluates this generation by measuring and computing the fitness function to reach its best fitness with the best-found final result, then it stopped and end (Figure 14).

However, although the GA outperformed the traditional Brute Force calibration method of traditional "SLEUTH" in a way to reduce the number of needed computational iterations and CPU run time and also to increase the modeling calibration speed, this improved performance still not yet reached its satisfactoriness. Therefore, coupling and combining in parallel two intelligent calibration methods such as

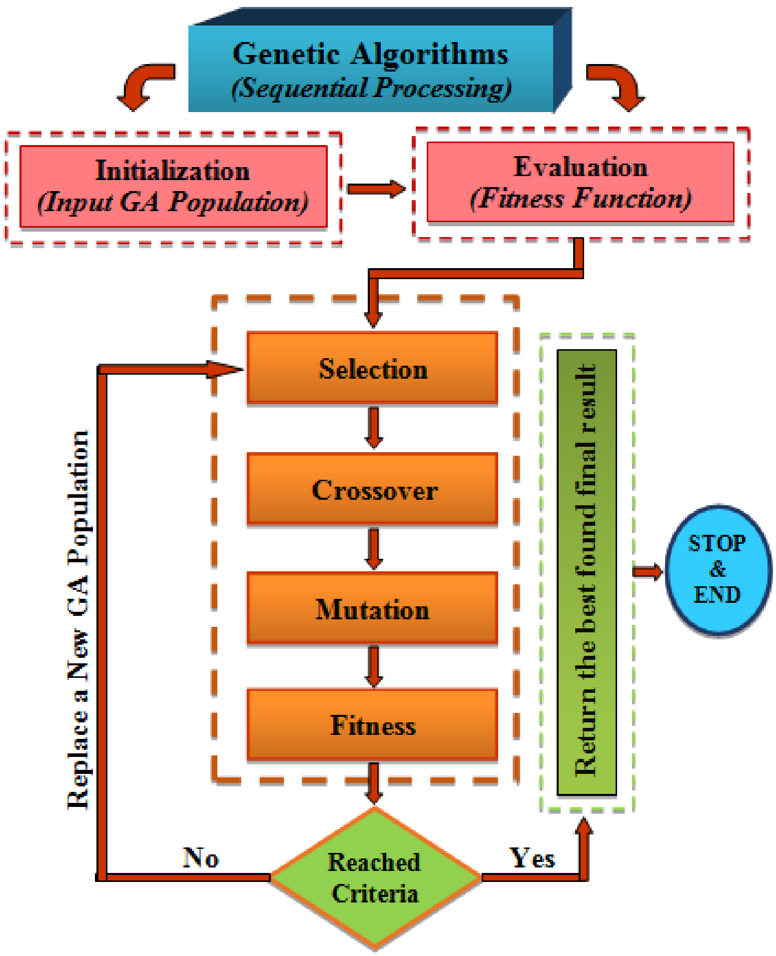

Figure 14. Sequential operational function of Genetic Algorithms (GA).

Machine Learning-GA and Deep Learning-CNNs is essential to obtain an optimal control parameters with the least achieved run time. The suggested coupling strategy is following the full integration one. For this purpose, the calibration module components of both GA and CNNs should use the same data structure and should share a common data management component, programming source code, and user interface. This full integration type of coupling strategy helps to facilitate the data exchange and management and it also reduce the calibration complexity of computing the SLEUTH's controlled coefficient parameters and reducing also the CPU run time of the model with less time and higher accuracy.

2.3 *Designing final block diagram of the modeling system:*

Figure 15 illustrate the final suggested block diagram that represents the full conceptual programming design of the "SLEUTH-CNNs" CA-based Urban Growth Modeling System. This complete software system has three main modes: processing, calibration, and prediction.

The processing mode preceded and started with preparing and developing the different data input. It is divided into two types of dataset categories. The first is the remotely sensed-based data, which is consists of a number of time series "historical and current date" satellite images (preferred at least three). These time series satellite images must fulfill several requirements and conditions, such as: all images must have the same spatial dimension, (spatial, spectral, satellite's temporal, and radiometric resolution), same coordinate system with same projection and datum, same anniversary date of each different time period image, and fixed time period between two satellite images (one as initial date and the other as final date; for example: 10 years' time difference between every two sequential images). These satellite images datasets are processed by a supervised Multi-Layer Perceptrons (MLP) of a CNNs-based Deep Learning Intelligent Classification Algorithm. The resultant land cover classes (LC) of each processed image are verified by measuring accuracy with loss function measurements. Urban built-up areas extent is extracted from every time period LC. The second data input category is all the other SLEUTH's required data. These time period datasets are: slope, different exclusion areas, transportation networks, and hillshade layers.

The calibration mode initialized the resultant processed input datasets by setting the main SLEUTH's five control parameters to measure the urban growth controlling required coefficients. These growth coefficient conditions are: Diffusion, Breed, Spread, Slope, and Road gravity influenced growth. The calibration mode in this research is a paralleled operational function of the full integrated CNNs-GA

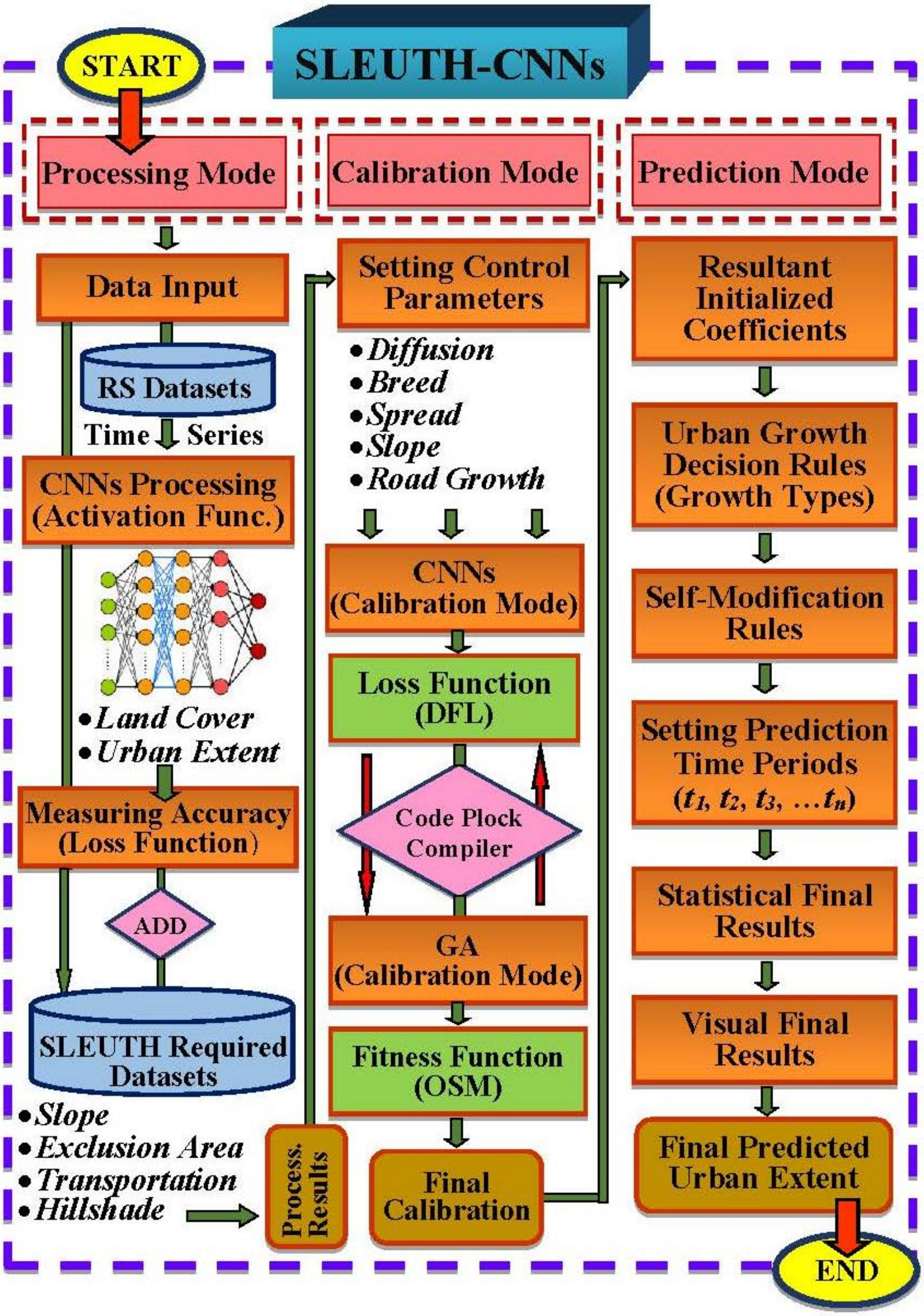

Figure 15. The Conceptual Modeling Final Design of the proposed "SLEUTH-CNNs" CA-based Urban Growth Modeling Software System, started with modeling processing mode and ending with prediction toward future mode.

suggested method. Loss and Fitness functions are measured by Distribution Focal Loss (DFL) and Optimal SLEUTH Metric (OSM), sequentially.

The final prediction mode is preceded with the resultant initialized growth controlling coefficients. From this point, the system would be able to generate the urban growth transition (decision) rules then self-modify and specify the growth type according to the pre-setting prediction desired period of time. The evolution of these transition rules leads to develop dynamical patterns that would be generated as a final dynamical visualization of future resultant urban extent along with developing a final statistical result just as well.

VIII. Discussion and Conclusion

This research is an attempt to propose a novel methodology to create a comprehensive system consisting of a set of GeoAI different algorithmic modeling techniques that work in parallel programming approach with each other to produce a new intelligent modified version of SLEUTH model. Noticeably that the integration between GeoAI methods and CA-based SLEUTH model is not a new attempt to be tried by several researchers or even by Clarke himself, but here in this research the integration has a greater scale. The full integration between these independently different operated and functioned modeling systems is converting the model from being a stand-alone simulative and predictive model to a fully operated individual intelligent multi-system computer software program. The created multi-integrated system contains many intelligent algorithmic methods that belong to "Bottom-Up" approach. In this respect, the proposed “SLEUTH-CNNs” model is designed to consist of three main mode pillars represented in the form of three fully integrated sub-systems. These modes are: processing, calibration, and prediction. There are two novelties in this research, one is concerned with the processing mode while the other is related to the calibration mode. The first novelty is started with processing a multi-layer perceptron-supervised deep learning CNNs (Convolutional Neural Networks) that accompanied with its internal convolution, pooling, activation, and loss algorithms. This whole first system part belongs to the artificial intelligent complex system theory. Then, comes another complex dynamic modeling algorithm such as Cellular Automata (CA) that was already integrated with Genetic Algorithms (GA) in its calibration part. Therefore, the second novelty's contribution is to expand the algorithmic integration approach of SLEUTH-GA's model to include the CNNs implementation mode to be fully integrated just as well with the Genetic Algorithms calibration part. The suggested integrated calibration mode preserves the basic controlling coefficients that are required by SLEUTH. Both Deep Learning based CNNs and Machine Learning based GA intelligent calibration methods are fully connected together in a parallel operational function along with their accuracy measurement methods of loss function (DFL) and fitness function (OSM), respectively. This procedure is reducing significantly the computational run time while obtaining the highest possible degree of accuracy. Moreover, this new suggested GeoAI-based modeling modification offers a new methodological perspective of coupling and fully integrating different types of geospatial intelligent methods. Finally, the proposed "SLEUTH-CNNs" modeling multi-system GeoAI approach that is presented in this research could be a promising new modification attempt for a well-known Urban Growth Model.

REFERENCES

1. Antoni, J-Ph, Vuidel, G., Omrani, H., & Klein, O., 2019, Geographic cellular automata for realistic urban form simulations: how far should the constraint be contained? *in*, D'Acci, L. (ed.), The Mathematics of Urban Morphology, Birkhäuser Basel; Springer Nature Switzerland AG, pp. 147-162.
2. Azaz, L.K., 2004, Monitoring, Modeling, and Managing Urban Growth in Alexandria, Egypt: Using Remote Sensing and GIS, Ph.D. Thesis, Geomatics Department, The University of Newcastle, U.K.
3. Batty, M., 1971, Modeling Cities as Dynamic Systems, *Nature*, vol. 231, pp. 425-428.
4. Batty, M., & Xie, Y., 1994, From cells to cities, *Environment and Planning B: Planning and Design*, 21, pp. 531-548.
5. Batty, M.; Xie, Y.; Sun, Z., 1999, Modeling urban dynamics through GIS-based Cellular Automata, *Computer, Environment and Urban Systems*, 23, pp. 205-233.
6. Batty, M., 2000, "GeoComputation using Cellular Automata", *in*, Openshaw, S., & Abrahart, R.J., (eds.), GeoComputation, Taylor and Francis, London, pp. 96-128.
7. Beyer, W.A.; Sellers, P.H.; Waterman, M.S., 1985, Stanislaw M. Ulam's contributions to theoretical theory, *Letter in Mathematical Physics*, 10, pp. 231-242.
8. Burks, A.W., (ed.), 1970, Cellular Automata, Univ. of Illinois Press, Urbana.
9. Chandan M.Ch., Aadithya J.S., Prakask P.S., & Aithal, B.H., 2019, Machine Learning for Building Extraction and Integration of Particle Swarm Optimization with SLEUTH for Urban Growth Pattern visualization for Livable Cities, *in*, Proceedings of the 55th ISOCARP-World Planning Congress-Beyond the Metropolis (9-13 September), pp. 54-74.
10. Chaudhuri, G., & Clarke, K.C., 2013, The SLEUTH Land Use Change Model: A Review, *The International Journal of Environmental Resources Research*, 1(1), pp. 88-104.
11. Chaudhuri, G., & Foley, S., 2019, DSLEUTH: A Distributed Version of SLEUTH Urban Growth Model, *Society for Modeling & Simulation International (SCS), SpringSim-ANSS*, April 29-May 2, Tucson, AZ, USA,

12. Clarke, K.C.; Brass, J.A.; Riggan, P.J., 1994, A Cellular Automaton Model of wildfire propagation and extinction, *Photogrammetric Engineering & Remote Sensing*, 60(11), pp. 1355-1367.
13. Clarke, K.C.; Gaydos, L.; Hoppen, S., 1997, A self modifying cellular automaton model of historical urbanization in the San Francisco Bay area, *Environment and Planning B: Planning and Design*, 24, pp.247-261.
14. Clarke, K.C., & Gaydos, L.J., 1998, Loose-Coupling a Cellular Automaton Model and GIS: Long-term urban growth prediction for San Francisco and Washington/Baltimore, *International Journal of Geographical Information Science*, 12(7), pp. 699-714.
15. Clarke, K.C., 2017, Improving SLEUTH Calibration with a Genetic Algorithm, *Proceedings of the 3rd International Conference on Geographical Information Systems Theory, Applications and Management (GISTAM)*, pp. 319-326.
16. Clarke, K.C., 2018, "Land Use Change Modeling with SLEUTH: Improving Calibration with a Genetic Algorithm", *in*, Olmedo, M., Paegelow, J., & *et al*., (Eds), Lecture Notes in Geoinformation and Cartography, Springer International Publishing, chap.8, pp. 139-162.
17. Clarke, K.C., & Johnson, J.M., 2020, Calibrating SLEUTH with big data: Projecting California's Land Use to 2100, *Computer, Environment and Urban Systems*, 83, pp. 1015-1025.
18. Clarke-Lauer, M.D., & Clarke, K.C., 2011, Evolving Simulation Modeling: Calibrating SLEUTH Using a Genetic Algorithm, Proceedings of the 11th International Conference on GeoComputation, University College London 20th-22nd July.
19. Cochinos, R., 2000, "Introduction to the theory of cellular automata and One-Dimensional traffic simulation", *in*, website: https://theory.org/complexity.
20. Couclelis, H., 1985, Cellular Worlds: a framework for modeling micro-macro dynamics, *Environment and Planning A*, 17, pp.585-596.
21. Couclelis, H., 1997, From cellular automata to urban models: new principles for model development and implementation, *Environment and Planning B: Planning and Design*, 24, pp.165-174.
22. Crecine, J. P., 1968, A Dynamic Model of Urban Structure, P-3803 (Rand Corporation, Santa Monica).

23. Deadman, P.D.; Brown, R.D.; Gimblett, H.R., 1993, Modelling rural residential settlement patterns with cellular automata, *Journal of Environmental Management*, 37, pp. 147-160.
24. Dietzel, C., & Clarke, K.C., 2007, Toward Optimal Calibration of the SLEUTH Land Use Change Model, *Transactions in GIS*, 11, pp. 29-45.
25. Eastman, J.R., 2016a, TerrSet: Geospatial Monitoring and Modeling System Manual, Clark University, U.S.A.
26. Eastman, J.R., 2016b, TerrSet: Geospatial Monitoring and Modeling System Tutorial, Clark University, U.S.A.
27. Eastman, J.R., & Toledano, J., 2018, A short presentation of the Land Change Modeler (LCM), *in*, Camach Olmedo, M., Paegelow, M., Mas, J-F., & Eacobar, F., (Eds), Geomatic Approaches for Modeling Land Change Scenarios, Springer, Cham, Switzerland, pp.499-505.
28. Elfwing, S., Uchibe, E., & Doya, K., 2017, Sigmoid-Weighted linear units for Neural Network Function Approximation in Reinforcement Learning, *Neural Networks*,107, pp. 3-11.
29. Esri.com, 2024, (https://www.esri.com/en-us/what-is-gis/history-of-gis).
30. Feng, Y., & Tong, X., 2019, A new cellular automata framework of urban growth modeling by incorporating statistical and heuristic methods, *International Journal of Geographical Information Science*, 34(1), pp. 74-97.
31. 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, pp. 1387–1400.
32. Ferreira, B.M., Soares-Filho, B.S., & Pereira, F., 2019, The Dinamica EGO virtual machine, *Science of Computer Programming*, 173, pp. 3-20.
33. Forrester, J., 1969, Urban Dynamics, MIT press, Cambridge, Massachusetts.
34. Gardner, M., 1970, The fantastic combinations of John Conway's new solitary game of life, *Scientific American*, 223, pp. 120-123.
35. Gigalopolis, 2003, Project Gigalopolis: Urban and Land Cover Modeling, The University of California-Santa Barbara (UCSB), *in* website: http://www.ncgia.ucsb.edu/projects/gig/.
36. Goldstein, N.C., 2004, Brains Vs. Brawn-Comparative strategies for the calibration of a Cellular Automata-Based Urban Growth Model, *in*, P.

Atkinson, G. Foody, S. Darby, & *et al.*, (Eds.), Geo Dynamics: CRC Press.
37. Green, D., 1993, Cellular automata tutorial, *in* website: http://life.csu.edu.au/complex/tutorials.
38. Guan, Q., & Clarke, K.C., 2010, A general-purpose parallel raster processing programming library test application using a geographic cellular automata model, *International Journal of Geographical Information Science*, 24, pp. 695-722.
39. Guan, D., Li, H., Inohae, T., Su, W., & et al., 2011, Modeling urban land use change by the integration of cellular automaton and Markov model, *Ecol. Model.*, 222, pp. 3761–3772.
40. Hewitt, R., Pacheco, J.D., & Gomez, B.M., 2013, A cellular automata land use model for the R software environment (Description and user documentation), SIMLANDER-User Guide.
41. Hewitt, R., 2015, APoLUS model full system documentation, Technical Report, pp. 1-33.
42. Hewitt, R., Roodposhti, M., & Bryan, B., 2022, There's no best model! Addressing limitations of land-use scenario modelling through multi-model ensembles, *International Journal of Geographical Information Science*, 36(12), pp. 2352-2385.
43. Hinton, G.E., Osindero, S., & The, Y.W., 2006, A fast learning algorithm for deep belief nets, Neural Computation, 18(7), pp. 1527-1554.
44. Hurd, L., 1987, Formal language characterizations of cellular automaton limit sets, *Complex Systems*, 1, pp. 69-80.
45. Jantz, C.A., Goetz, S.J., Donato, D., & Claggett, P.R, 2010, Designing and Implementing a Regional Urban Modeling System Using the SLEUTH Cellular Urban Model, *Computers, Environment and Urban Systems*, 34(1), pp. 1-16.
46. Kocabas, V. & Dragicevic, S., 2007, Enhancing a GIS cellular automata model of land use change: Bayesian networks, influence diagrams and causality, *Trans. GIS*, 11, pp. 681–702.
47. LeCun, Y., Boser, B., Denker, J.S., Henderson, D., & *et al.*, 1989, Backpropagation applied to handwritten zip code recognition, *Neural Computation*, 1(4), pp. 541-551.

48. LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P., 1998, Gradient-based learning applied to document recognition, *Proceedings of the IEEE*, 86(11), pp. 2278-2324.
49. Lee, H., Grosse, R., Ranganath, R., & Ng, A.Y., 2009, Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations, *in*, ICML, pp. 609-616.
50. Li, B., Zheng, Z., & Zhang, C., 2021, Identifying Incorrect Classifications with Balanced Uncertainty, (https://arxiv.org/ pdf/2110.08030).
51. Li, Xia., & Yeh, Anthony Gar-On., 2002, Neural-network-based cellular automata for simulating multiple land use changes using GIS, *International Journal of Geographical Information Science*, 16(4), pp. 323-343.
52. Li, Xia., & Yeh, Anthony Gar-On., 2004, Data mining of cellular automata's transition rules, *International Journal of Geographical Information Science*, 18(8), pp. 723-744.
53. Liu, X., Liang, X., Li, X., Xu, X., & *et al*., A future land use simulation model (FLUS) for simulating multiple land use scenarios by coupling human and natural effects, *Landscape and Urban Planning*, 168, pp. 94-116.
54. Liu, Y., & Phinn, S.R., 2003, Modelling urban development with cellular automata incorporating fuzzy-set approaches, *Computers, Environment and Urban Systems*, 27, pp. 637-658.
55. Naghibi, F.; Delavar, M.; and Pijanowski, B., 2016, Urban growth modeling using cellular automata with multi-temporal remote sensing images calibrated by the artificial bee colony optimization algorithm, *Sensors*, 16, pp. 1-21.
56. Najarro, E.; Sudhakaran, S.; Glanois, C.; and Risi, S., 2022, HyperNCA: Growing development networks with neural cellular automata, *International Conference on Learning Representations Workshops (ICLR Workshops)*, pp.1-11.
57. Penatti, O.A., Nogueira, K., & Dos Santos, J.A., 2015, Do Deep Features Generalize from Everyday Objects to Remote Sensing and Aerial Scenes Domains? Workshop paper presented at the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 44-51.
58. RIKS BV, 1992, METRONAMICA Documentation, The Netherlands.
59. Schatten, A., 2018, "Cellular Automata: Digital Worlds", *in*, website: *http://www. schatten.info/resources_ca.html*.

60. Shan, J., Alkheder, S., & Wang, J., 2008, Genetic Algorithms for the Calibration of Cellular Automata Urban Growth Modeling, *Photogrammetric Engineering & Remote Sensing* (*PE&RS*), 74 (10), pp. 1267-1277.
61. Silva, E.A., & Clarke, K.C., 2002, Calibration of the SLEUTH Urban Growth Model for Lisbon and Porto, Portugal, *Computers, Environment and Urban Systems*, 26, pp. 525-552.
62. Sipahioglu, N., & Cagdas, 2023, Scenario-Based Cellular Automata and Artificial Neural Networks in Urban Growth *Modeling, GU J Sci*, 36(1), pp.20-37.
63. Stanilov, K., & Batty, M., 2011, Exploring the historical determinants of urban growth patterns through cellular automata, *Transactions in GIS*, 15(3), pp.253-271.
64. Takeyama, M. & Couclelis, H, 1997, Map dynamics: integrating cellular automata and GIS through Geo-Algebra. *International Journal of Geographical Information Systems*, 11(1), pp. 73-91.
65. Tesfaldet, M.; Nowrouzezahrai, D.; and Pal, C., 2022, Attention-based neural cellular automata, *36th Conference on Neural Information Processing Systems (NeurLPS)*, 35, pp. 8174–8186.
66. Tobler, W., 1979, "Cellular Geography", *in*, Gale, S. & Olsson, G., (eds.), Philosophy in Geography, Riedel, Dortrech, pp. 379-386.
67. Toffoli, T., & Margolus, N., 1987, Cellular Automata: A new environment for modeling, Cambridge, MA: MIT press, U.K.
68. Torrens, P.M., 2003, "Cellular Automata and multi-agent systems as planning support tools", *in*, Geertman, S., & Stillwell, J., (Eds.), Planning support systems in practice, Springer-Verlag Berlin Heidelberg, New York, Chap.pp. 205-222.
69. Torrens, P.M., 2011, "Calibrating and Validating Cellular Automata Models of Urbanization", *in*, Yang, X., (ed.), Urban Remote Sensing: Monitoring, Synthesis and Modeling in the Urban Environment, Wiley & Sons, Chichester, Chap.pp. 335-345.
70. Torrens, P.M., 2012, "Building Mega-Models for Megacities", *in*, De Roo, G., Hillier, J., & Van Wezemael, J., (eds.), Complexity and Planning: Systems, Assemblages and Simulations, Ashgate Publishing Limited, UK, Chap.pp. 409-425.

71. Veldkamp, A., & Fresco, L.O., 1996, CLUE: a conceptual model to study the conversion of land use and its effects, *Ecological Modelling*, 85, pp.253-270.
72. Verburg, P., Soepboer, W., Veldkamp, A., Espaldon, M., & *et al*., 2002, *Environmental Management*, 30(3), pp. 391-405.
73. Verburg, P., & Overmars, K., 2009, Combining top-down and bottom-up dynamics in land use modeling: Exploring the future of abandoned farmlands in Europe with the Dyna-CLUE model, *Landscape Ecology*, 24(9), pp. 1167-1181.
74. Vakalopoulou, M., Christodoulidis, S., Burgos, N., Colliot, O., & *et al*., 2023, Deep Learning: Basics and Convolutional Neural Networks (CNNs), *in*, Colliot, O., (ed.), Machine Learning for Brain Disorders; Springer, Chap.3, pp. 77-115.
75. White, R., & Engelen, G., 1993, Cellular automata and fractal urban form: a cellular modelling approach to the evolution of urban land use patterns, *Environment and Planning A*, 25, pp. 1175-1199.
76. White, R., & Engelen, G., 1997, Cellular automata as the basis of integrated dynamic regional modeling, *Environment and Planning B: Planning and Design*, 24, pp. 235-246.
77. White, R., Engelen, G., & Uljee, I., 1997, The use of constrained cellular automata for high-resolution modelling of urban land-use dynamics, *Environment and Planning B: Planning and Design*, 24, pp. 323-343.
78. White, R., & Engelen, G., 2000, High-resolution integrated modeling of the spatial dynamics of urban and regional systems, *Computers, Environment and Urban Systems*, 24(5), pp. 383-400.
79. Wolfram, S., 1984, Cellular Automata as models of Complexity, *Nature*, 311(5985), pp. 419-424.
80. Wu, F., 1998, An experiment on the generic polycentricity of urban growth in a cellular automatic city, *Environment and Planning B: Planning and Design*, 25, pp. 103-126.
81. www.sciencebase.gov., 2019.
82. Xu, H.Q.; Yang, H.C.; Zhi Feng L.; Pei Jun S., 2014, Modeling the impacts of drying trend scenarios on land systems in northern China using an integrated SD and CA model. *Sci. China Earth Sciences*, 57(4), pp. 839-854.
83. Yadav, N., Yadav, A. & Kumar, M., 2015, An Introduction to Neural Network Methods for Differential Equations, Springer Briefs in Applied

Sciences and Technology-Computational Intelligence, Springer Dordrecht Heidelberg, New York.

84. Yang, X., & Lo, C.P., 2003, Modeling urban growth and landscape changes in the Atlanta metropolitan area. *International Journal of Geographical Information Science, 17*, pp. 463-488.
85. Zare, M.; Panagopoulos, T.; Loures, L., 2017, Simulating the impacts of future land use change on soil erosion in the Kasilian watershed, Iran, *Land Use Policy*, 67, pp. 558–572.
86. Zhou, y., Varquez, A.C.G., & Kanda, M., 2019, High-resolution global urban growth projection based on multiple applications of the SLEUTH urban growth model, Nature: Scientific Data, 6 (34), pp. 1-10.
87. Zhou, Z., & Hu, P., 2016, GIS and Coastal Basemap Research Summer Internship with Clark Labs, *International Development, Community and Environment (IDCE)*, 79, pp.1-34.