FEDERAL UNIVERSITY OF MINAS GERAIS

School of Engineering

Graduate Program in Electrical Engineering

Lucas Malacarne Astore

Data-driven spatio-temporal modeling with Cellular Automata and Fuzzy Time Series methods

Belo Horizonte

2022

Lucas Malacarne Astore

Data-driven spatio-temporal modeling with Cellular Automata and Fuzzy Time Series methods

Final version

Master thesis presented to the Graduate Program in Electrical Engineering at Federal University of Minas Gerais in partial fulfillment of the requirements for the degree of Master of Science in Electrical Engineering

Supervisor: Frederico Gadelha Guimarães

Co-supervisor: Carlos Alberto Severiano Júnior

Belo Horizonte
2022

Astore, Lucas Malacarne.
A858d Data-driven spatio-temporal modeling with cellular automata and Fuzzy Time Series methods [recurso eletrônico] / Lucas Malacarne Astore. - 2022.
1 recurso online (86 f. : il., color.) : pdf.

Orientador: Frederico Gadelha Guimarães.
Coorientador: Carlos Alberto Severiano Júnior.

Dissertação (mestrado) - Universidade Federal de Minas Gerais, Escola de Engenharia.

Bibliografia: f. 77-86.
Exigências do sistema: Adobe Acrobat Reader.

1. Engenharia elétrica - Teses. 2. Aprendizado do computador - Teses. 3. Modelagem - Teses. I. Guimarães, Frederico Gadelha. II. Severiano Júnior , Carlos Alberto. III. Universidade Federal de Minas Gerais. Escola de Engenharia. IV. Título.

CDU: 621.3(043)

Ficha catalográfica elaborada pela bibliotecária Ângela Cristina Silva CRB/6 2361
Biblioteca Prof. Mário Werneck, Escola de Engenharia da UFMG

UNIVERSIDADE FEDERAL DE MINAS GERAIS
ESCOLA DE ENGENHARIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

FOLHA DE APROVAÇÃO

"DATA-DRIVEN SPATIO-TEMPORAL MODELING WITH CELLULAR AUTOMATA AND FUZZY TIME SERIES METHODS"

LUCAS MALACARNE ASTORE

Dissertação de Mestrado submetida à Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em Engenharia Elétrica da Escola de Engenharia da Universidade Federal de Minas Gerais, como requisito para obtenção do grau de Mestre em Engenharia Elétrica. Aprovada em 16 de dezembro de 2022. Por:

Prof. Dr. Frederico Gadelha Guimarães
DEE (UFMG) - Orientador

Prof. Dr. Carlos Alberto Severiano Junior
(IFMG-Sabará) - Coorientador

Prof. Dr. Tiago Garcia de Senna Carneiro
DECOM (UFOP)

Prof. Dr. Pedro Paulo Balbi de Oliveira
FCI (Universidade Presbiteriana Mackenzie)

Prof. Dr. Walmir Matos Caminhas
DELT (UFMG)

Documento assinado eletronicamente por **Frederico Gadelha Guimaraes**, **Coordenador(a) de curso de pós-graduação**, em 19/12/2022, às 09:40, conforme horário oficial de Brasília, com fundamento no art. 5º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Tiago Garcia de Senna Carneiro**, **Usuário Externo**, em 19/12/2022, às 17:56, conforme horário oficial de Brasília, com fundamento no art. 5º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Walmir Matos Caminhas**, **Professor do Magistério Superior**, em 20/12/2022, às 10:34, conforme horário oficial de Brasília, com fundamento no art. 5º

do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Carlos Alberto Severiano Junior**, **Usuário Externo**, em 21/12/2022, às 09:13, conforme horário oficial de Brasília, com fundamento no art. 5º do Decreto nº 10.543, de 13 de novembro de 2020.

Documento assinado eletronicamente por **Pedro Paulo Balbi de Oliveira**, **Usuário Externo**, em 21/12/2022, às 15:19, conforme horário oficial de Brasília, com fundamento no art. 5º do Decreto nº 10.543, de 13 de novembro de 2020.

A autenticidade deste documento pode ser conferida no site https://sei.ufmg.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **1958837** e o código CRC **F8C9F33B**.

To Mom and Dad

Acknowledgements

Among the challenges in pursuing a master's degree, I had to face all the fear, anxiety, and uncertainties that the pandemic of Covid-19 brought in the past years, during the development of this work. It was indeed an intense and difficult period of my life and I'm grateful and fulfilled to close this chapter.

First, huge thanks to my advisor Professor Frederico G. Guimarães, and my co-advisor Professor Carlos A. Junior. This dissertation wouldn't be possible at all without the meetings, brilliant insights, and guidance I've received from them. They gave me all the support and help I needed to develop this research, even from distance during the lockdown.

I'm always grateful to my mother Aparecida Malacarne, for all the support, caring, and love, and to my father Jayme Astore for his effort along my education path. Also thanks to my brother Matheus Malacarne, for the special moments shared with my cats Dio, Fred, Théo, and Mila (RIP) for the emotional support. I love you all deeply.

Big thanks to my dear friends family Cristina Marchiori and Marcello Dório who shared with me daily life special and difficult moments. Both of you were really important to me during these years. Thank you, dear Rafael Vulpi, for the support, the laughs, and for being there with me during the scariest time. Thanks to my recent friends for sharing amazing moments this year, Thaís Rabello, Lívia Valle, Thaís Caldeira, and Affonso Favalessa, and my friends from high school/life: Gabriela Gomes, Matheus Bayer, Marcella Freisleben, Thaísa Caroline, Rayane Venturini, Lívia Monteiro and Samarony Oliveira for always sending me positive energy to move forwards.

Finally but not less importantly, thank you CAPES for funding my studies and the postgraduate program of the Federal University of Minas Gerais for the opportunity. I will always support and advocate for the right of all people to access high-quality public education.

"Looking at the past must only be a means of understanding more clearly what and who they are - so that they can more wisely build the future." Paulo Freire

Resumo

Há atualmente diversas aplicações de simulações computacionais em estudos de sistemas dinâmicos espaço-temporais, incluindo, por exemplo, em modelos epidemiológicos. Dentre as estratégias capazes de reproduzir e predizer o futuro dos estados e comportamentos dinâmicos, os autômatos celulares (em inglês, cellular automata - CAs) são frequentemente aplicados na modelagem espaço-temporal. O conceito central de um típicoe bem definido modelo CAs é o desenvolvimento de um conjunto de regras locais que descrevem os estados futuros das células considerando as células vizinhas. O processo de construção deste conjunto exige conhecimento técnico e anos de pesquisa científica. Técnicas baseadas em aprendizado de máquina podem ser aplicadas para automatizá-lo, embora sejam necessários algoritmos de otimização de hiperparâmetros. Nesse contexto, este trabalho apresenta uma abordagem orientada a dados para definição de conjuntos de regras de transição de CA, baseada exclusivamente em dados históricos de um determinado fenômeno espaço-temporal. As regras locais do autômato são aprendidas e representadas usando o método Multivariado Fuzzy Time Series (MFTS). O modelo MFTS é então integrado à simulação do CA, funcionando de forma semelhante a um conjunto tradicional de regras. A metodologia proposta foi testada em dois casos de estudo: Espalhamento Espacial da Doença de Chagas e Dinâmica da Mudança do Uso e Cobertura do Solo em Delhi, na Índia. Em ambos conjuntos de dados, verificou-se grande potencial no uso do modelo FTS como estratégia de transição de estados em CA.

Palavras-chave: Fuzzy Times Series, Cellular Automata, Modelagem espaço-temporal, Uso e Cobertura de Territórios, Modelagem dinâmica.

Abstract

There have been several applications of computer simulations in studies of spatio-temporal dynamic systems, including epidemiological models. Among the strategies capable of reproducing and predicting future states and behaviors over time, Cellular Automata (CAs) are often applied in geospatial environmental modeling. The core concept of a typical and well-defined CAs model is the development of local rules set that describe the future cell states considering the neighboring cells. The process of building this set demands technical knowledge and years of scientific research. Machine learning-based techniques can be applied in order to automate it, although hyper-parameter optimization algorithms are required. Therefore, this work presents a data-driven approach for CA transitional rules set definition, based exclusively on historical data of a given spatio-temporal phenomenon. The local rules of the automaton are learned and represented using a Multivariate Fuzzy Time Series (MVFTS) method. The MVFTS model is then integrated into the CA simulation, working similarly to a traditional set of CA rules. The proposed methodology was tested using two study cases: Spatial Spread of Chagas Disease and Land Cover/Use Change in Delhi, India. In both sets of data, there was great potential for using the FTS model as a state transition strategy in CA.

Keywords: Fuzzy Times Series, Cellular Automata, Spatio-temporal modeling, Land Cover Land Usage, Dynamics modeling.

List of Figures

List of Tables

List of abbreviations and acronyms

ABNT	Associação Brasileira de Normas Técnicas
MINDS	Machine Intelligence and Data Science
UFMG	Universidade Federal de Minas Gerais
FTS	Fuzzy Times Series
CA	Cellular Automata
FCA	Fuzzy Cellular Automata
DCA	Deterministic Cellular Automata
PCA	Probability Cellular Automata
LUCC	Land Use and Cover Change
GIS	Geographical Information Systems
SIR	Susceptible Infected Removed
TSF	Time Series Forecasting
FLR	Fuzzy Logic Relationship
FLRG	Fuzzy Logical Relationship Group
LHS	Left-Hand Side
RHS	Right-Hand Side
e-MVFTS	Evolving Multivariate Fuzzy Times Series
WFLRG	Weighted Fuzzy Logical Relationship Group
RSME	Root Mean Squared Error
nRSEM	Normalized Root Mean Squared Error
MAE	Mean Absolute Error

Contents

Chapter 1

Introduction

Computer simulations have been widely used to reproduce dynamical systems, in order to analyze, evaluate and predict real-world spatio-temporal scenarios Parker et al. [2003]. The knowledge extracted from those models can be used as a tool in the decision-making process, especially when planning effective and efficient public policies Turner Ii et al. [1995]. Global environmental challenges, such as deforestation, rapid population growth, pandemics, and global warming are examples of those complex spatio-temporal systems.

Building a modeling strategy for a spatio-temporal dynamical system demands identifying relevant human and natural factors, that affect the behavior in space and reproduce future spatial events. In wildfires, for instance, wind speed, type of forest, distance to water, and urban area might be relevant aspects to simulate the propagation of the fire over time under a region Zheng et al. [2017]. The complexity of a model is associated with the number of interactions and correlations the system variables may have Song [1999]. By changing those variables and factors it is also possible to measure and analyze public policy and its influence and social and economic impacts over a period of time under specific conditions Carneiro [2006]. For example, in compartmental models in epidemiology, simulations are made in order to evaluate the impacts of exposure and propagation of the disease, considering infection rates and an initial state. Those models are known as Susceptible-Infected-Recovered (SIR) Hethcote [2000], and can be used as tools to study lockdown and interventions, as recently seen during the Coronavirus Disease-2019 (COVID-19) pandemic research, in many regions of the world Talukder [2020], Din and Algehyne [2021], Brugnano et al. [2021].

Capturing spatial interactions and their dynamical patterns, involves dealing with multiple variables and parameters, as already mentioned. In this sense, it is relevant to note that the advances in data storage capacity along with the improvement of processing and analysis techniques play a crucial role in this field. In terms of Land Use/Cover Change (LUCC) simulations, as Diniz [2021] describes, spatio-temporal data can be obtained

from Geographic Information System (GIS), satellite or unmanned aerial vehicle spatial images, on-site sensors measurements, and social-economics information. Thus, LUCC modeling requires techniques capable to deal with different types of data, as recent Machine learning-based approaches Wang et al. [2022]. An accurate knowledge extraction process has the potential to assist in the categorization and forecasting of different events.

In the past decades, as reviewed in Ghosh et al. [2017], cellular automata (CAs) and Markov chain models represent two outstanding methods in geographic and spatial applications. A cellular automaton (CA) can be defined as a mathematical abstraction of a dynamical system in a discrete world. It consists of a regular lattice of machines and a finite set of discrete states which can change over time conditioned by a set of local rules or transitional rules. Each cell in the CA lattice is updated locally leading to complex global behaviors Toffoli and Margolus [1987]. In other words, the CA computational model allows the simulation of complex macroscopic dynamical systems based on local interactions, this being one of the key reasons to be chosen in many spatio-temporal applications and more specifically in LUCC models. The CAs strategy aims to be an alternative to differential equations, which usually is a highly time-consuming path Chopard and Droz [1998]. CA-based models are found in a large variety of study areas, such as deforestation process Carneiro [2006], urban sprawl Aburas et al. [2016] and spread of diseases such as the pandemic of COVID-19 Schimit [2021], Lima and Balbi [2022], Chagas Slimi et al. [2009] and Dengue fever Áurea Tonetti Massahud [2011]. Furthermore, CAs are flexible and often integrated into systems, composing hybrid and multi-agent system models Parker et al. [2003]. Fuzzy theory by Zadeh [1965] was firstly integrated by Cattaneo et al. [1997] defining fuzzy states, instead of a finite Boolean configuration, in the transitional function obtaining complex dynamics of the shifting rules. Since then, many researchers have been using the strategy, for example in epidemiological models Costa et al. [2013] or in studies of neighboring cell interaction Basu and Basu [2008].

A central concept of a typical and well-defined CA model is building a proper set of transition rules which describes the future cell states considering the neighboring cells Kari [2005]. Defining this set of rules may require expert and technical knowledge and years of scientific research which is time-consuming and it has a significant effort. Methods have been proposed in order to overcome this issue by using Artificial Neural Networks (ANN) and Data-mining techniques. Although the integration CA-ANNs strategy has shown effective at simulating non-linear relations among features, it consists of a Black-box method, it often demands hyper-parameters optimization, such as neurons configurations in input, hidden and output layers, and it has faced issues dealing with time series images, extremely common in LUCC modeling Wang et al. [2022], Qiang and Lam [2015]. The See5 system Data-mining tool was proposed by Li and Yeh [2004] to automatically generate transition rules. It consists of a decision-tree learning algorithm, based on the optimization of a gain function, able to deal with GIS data. The Overall accuracy obtained was on

average 75%, and no fuzzy relations were covered in the study case.

As mentioned, data storage systems together with recent remote sensing technology allow the composition of a large amount of time series information (Big Data) used as input in data-driven models. More specifically, spatio-temporal forecasting represents a class of methods in which nearby areas affect local changes over time Severiano et al. [2021]. Traditional statistics are available for probabilistic and interval forecasting, as instance Auto-Regressive Integrated Moving Average (ARIMA) Wabomba [2016], Bayesian Structural Time Series (BSTS), k-Nearest Neighbors (k-NN), among others. Nevertheless, Machine learning and soft computing methods offer golden tools to deal with spatial image analysis challenges, using for example neural networks Pijanowski et al. [2002], Gharaibeh et al. [2020].

In this context, the fuzzy time series (FTS) technique, primarily elaborated by Song and Chissom [1993], builds a forecasting model based on a fuzzy representation of conventional time series data. It is a data-driven method that allows high interpretability, since it uses fuzzy rules given by *IF-THEN* statements, and flexibility, working on multiple data types. FTS methods can consider monovariate or multivariate scenarios and weighted or weightless systems. Selecting the appropriate feature is essential to build a well-defined FTS model. Since its conceptualization, many improvements can be found, such as Trend WFTS Cheng et al. [2006], Improved WFTS Ismail and Efendi [2011], Exponentially WFTS Suhartono et al. [2011] and Probabilistic WFTS de Lima Silva et al. [2020]. In Severiano et al. [2021], it was developed the Evolving Multivariate FTS (e-MVFTS) method which associates spatio-temporal data with the FTS model in forecasting renewable energy, although it was not applied to a cellular automaton representation. FTS applications were employed in a myriad of studies such as seasonal time series Song [1999], stock index prices Huarng and Yu [2005], electric load Efendi et al. [2015] and others Bose and Mali [2019].

Considering the exposed scenarios, a strategy is proposed to incorporate FTS methods to learn dynamical patterns and generate the transitional rules of a given cellular automaton, fully based on a historical dataset. Since it consists of a supervised learning algorithm, it is divided into training and forecasting procedures. Essentially, a training dataset is used to build the FTS model, and then it is applied to a cellular automaton simulation, given a condition at time (t), predicting CA lattice scenario at $(t+1)$. The current study represents one of the first attempts in the literature to integrate fuzzy time series and cellular automata (CA-FTS) techniques. As proof of concept, two study cases were used for validation: an epidemiological model for Chagas disease spread and urban sprawl LUCC model in Delhi, India. Evaluation metrics were used to compare the test data and the output obtained from the CA-FTS model and it has shown high accuracy rates for both cases, representing a great potential of the proposed methodology in CA-based models.

1.1 Objectives

The main goal of this work is to present a new approach for building a set of transitional cellular automaton rules based on fuzzy time series methods. As exposed, local rules definition in CA is not a trivial task, and, in this sense, the CA-FTS model is an alternative to automatically define it from the historical data of a phenomenon, avoiding the usage of formulation-tailored rules and calibrating specific parameters. Specific goals can be described as:

- Study different approaches for cellular automaton modeling and hybrids methods with Machine Learning techniques;
- Present and develop an alternative computational model strategy to incorporate a FTS model, based on historical spatio-temporal data, to automatically determine the transitional rules component in a deterministic cellular automaton simulation;
- Present the mathematical foundation of FTS methods which allows its integration with CAs models;
- Apply the proposed methodology into study cases in order to demonstrate its applicability and evaluate the selected evaluation metrics;
- Discuss the implications and analyze the output obtained from the CA-FTS models' application and possible variations of its construction;

1.2 Dissertation Outline

The current work is organized as follows:

- Chapter 2 presents a theory overview of cellular automaton, fuzzy times series methods, and related work detailed;
- Chapter 3 introduces the methodology used in order to achieve the specified goals. The mathematical support of the CA-FTS integration is demonstrated and discussed;
- Chapter 4 described the details and the universe of both datasets used to validate the proposed method;
- Chapter 5 presents the results and the evaluation metrics obtained from the comparison of test data and CA-FTS model output. Lastly, a discussion wraps up the implications.
- Chapter 6 describes the main conclusions and a brief discussion about future work paths are analyzed.

Chapter 2

Theory Overview

2.1 Cellular automata

The computational model concept of cellular automaton (CA) was first initiated by John Von Neumann, in 1947 Neumann and Burks [1966]. Neumann's initial motivation was to model the biological system and its ability to self-reproduce and self-repair and apply similar properties to electronic circuits Chopard and Droz [1998]. The exhaustive work dealing with mathematical formalism in continuous time and space was solved following the suggestions of Stanislaw Ulam, adopting a discrete system composed of interconnected finite-state lattice machines. In the proposal, each machine has a state that can be changed according to the states of neighboring machines Kari [2005]. Thus, an evolutionary system was developed, capable of replicating such states from a configuration of an initial machine's states and a set of rules for determining neighborhood states, in discrete time and space. **The importance of Neumann's work relies on the idea of a discrete and finite automaton model capable of building global complex behaviors based on simple local rules set.**

After the conceptualization, a variety of spatio-temporal problems were simulated through cellular automata (CAs) models in order to understand the dynamics and the global behavior of phenomena. In Conway [1970] was proposed a conventional two-dimensional CA system method, known as the Game of Life, where the set of cells' states was composed of two possible elements: alive and dead. The system was based on very simple rules that lead to a complex dynamical behavior over time Gardner [1970]. In the 1980s, Stephen Wolfram studied one-dimensional CA rules and defined the concept of an elementary cellular automaton, the simplest temporal evolution model capable to generate a complex system Wolfram [1984]. His contribution was relevant once it proved that CAs models are essential objects to consider for statistical mechanics studies.

CA-based modeling is considered an alternative to dealing with differential equations when simulating and studying dynamical systems, given its simplicity and inter-

pretability Santé et al. [2010]. Furthermore, this class of models is very flexible, as it can easily adapt to different types of real-world scenarios. In terms of the model configuration, it has an open structure, allowing being used coupled with other methods Clarke et al. [1997] Aburas et al. [2016]. These characteristics are relevant in terms of LUCC studies because it usually requires integration with Geographical Information Systems (GIS) type of data with different levels of spatial resolution. Although broadly used, one of the challenges in LUCC CA-based modeling is the lack of not incorporating social-economic and non-geospatial factors Sang et al. [2011] Deep and Saklani [2014].

In the following sections, the cellular automaton method is defined, and its different types of structure and neighborhood configuration are presented. Then the transitional set of rules is detailed with an illustration of a classic application of epidemiological models. And finally the concept of boundary condition in CA computational simulation.

2.1.1 Definition

A cellular automaton is a modeling framework of dynamical processes, or a mathematical abstraction of the real world in a discrete universe – time, space, and states. It consists of a structure of spatially grouped cells, known as a lattice of cells, which evolve from state to time. Cell states evolution is conditioned by a set of transition rules, without a central processing unit. The cell future state at the time (t+1) is a function of the previous states of the surrounded cells and its own at time (t) Chopard and Droz [1998], Oliveira et al. [2001]. As described in Kari [2005], CA definition can be specified as:

1. A d-dimensional rectangular grid of squared cells, or lattice of cells, addressed by $\mathbb{Z}^d$. Commonly cases are one-, two- and three-dimensional grids;

2. A finite state set $\mathcal{S}$, mapping $c : \mathbb{Z}^d \to \mathcal{S}$ at a given time specifying the state of all cells. The set $\mathcal{S}^{\mathbb{Z}^d}$ defines all the possible configurations and is also known as $\mathbb{C}$;

3. The discrete time step t defines an updating cycle of all the cells in the lattice, accordingly to the states of the neighboring cells, the central cell, and a local rule. In terms of real-world applications, for example, t can be considered a day, a month, three months, a year, and so forth.

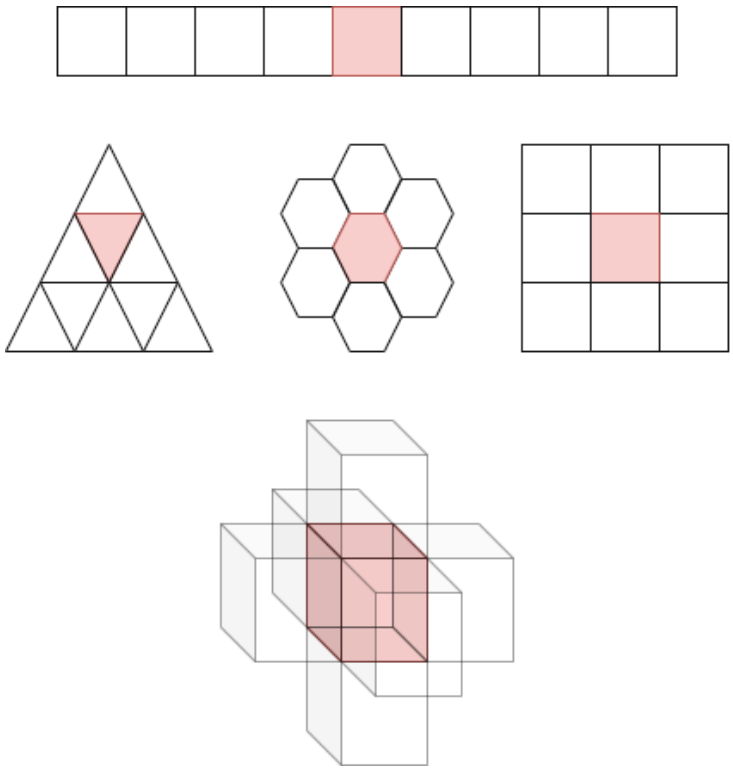
4. Neighborhood vector N definition, where $N = (\vec{x_1}, \vec{x_2}, ..., \vec{x_n})$, for n distinct elements of $\mathbb{Z}^d$. The neighbors of a cell at location $\vec{x} \in \mathbb{Z}^d$ are the cells at location $\vec{x} + \vec{x_i}$ for $i = 1, 2, ..., n$;

5. A function $f : \mathcal{S}^n \to \mathcal{S}$, where n is the size of the neighborhood. The state at time t, $y_c = f(y'_1, y'_2, ..., y'_n)$, is the state of a cell c whose the n neighbors were at states $y'_1, y'_2, ..., y'_n$ at time $(t - 1)$. This transitional or update rule determines the global dynamics of the CA.

2.1.2 Structure and Neighborhood

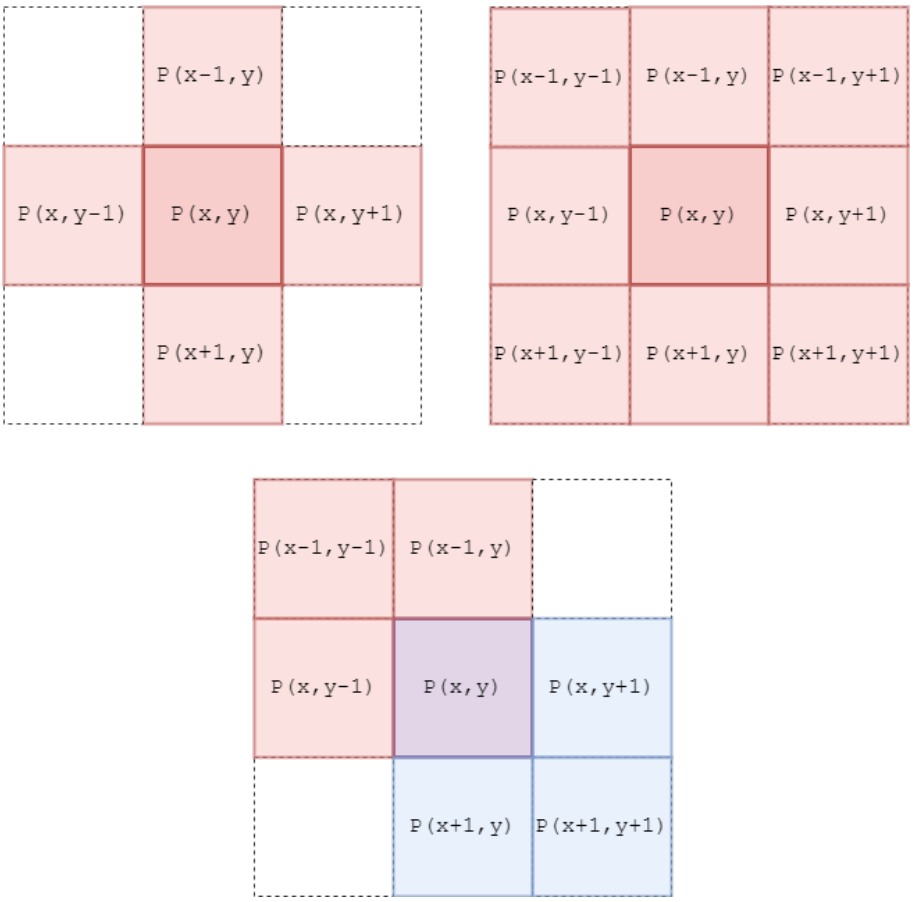
The CA cells can be simulated considering different types of shapes. The group of interconnected cells defines a lattice of a CA system. In terms of dimension, CA can vary accordingly to the model of interest. Initially most of the work considered one-dimensional (1D) and two-dimensional (2D) CAs, but they can also have three-dimensional (3D) structures. Figure 1 presents possible configurations applicable to CA-based models.

Figure 1 – Different configurations of a CA



As mentioned in the definition, the future state of a cell is predicted by the current cell state and its vicinity states. These nearby cells are called the neighborhood and they can be modeled in a variety of sizes and ways. The size of a neighborhood, also known as the radius ($r \in \mathbb{N}$), is the distance from the central cell, and it is proportional to the complexity of the model, in other words, the larger is the radius the higher is the CA complexity Chopard and Droz [1998]. For a 2D CA, the most traditional ones, considering radius 1, are Von Neumann, in which the four cells to the north, south, east, and west of the target cell are considered neighbors, and Moore which considers all cells around the target one. Another well-known and defined neighborhood is known as Margolus, Toffoli and Margolus [1987]. The neighborhood is divided into blocks of four cells and the blocks alternate at each time step, covering a larger space outside the traditional boundary. This approach allows a reduction of rule complexity in relation to simply increasing the radius. Figure 2 shows the arrangement of the aforementioned neighborhoods, where $P(x, y)$ is an element of a matrix of discrete cells Ghosh et al. [2017].

Figure 2 – Three different neighborhood configurations: Von Neumann, Moore, and Margolus.



In contrast to the standard neighborhood, it is worth mentioning that there are several other types of neighborhood dispositions since it is a configuration that can be adjusted according to the system to be modeled Kari [2005]. For example, in Schimit [2021], the authors propose specific neighborhood dynamical simulations for CA SIR models, where, at the beginning of each time step, individuals make C connections to neighbors inside a Moore radius r.

2.1.3 Transition rules

The future cell's states are locally determined based on its actual state, the states of the neighbors, and a set of transitional rules. This is the key to the simulation of dynamic events. Initially, transitional rules of CA models have one single possible state output, i.e. they were deterministic (Deterministic Cellular Automata - DCA) Ghosh et al. [2017]. In order to model the vagueness and uncertainties of the real-world scenarios, a probability was associated with the local rules. In Probabilistic Cellular Automata (PCA) the updating rule is driven by external probabilities. In general, real-world phenomena are probabilistic and their values and distribution are defined for each phenomenon, some examples of applications are in modeling epidemiological systems such as hepatitis B infections Xiao

et al. [2006], Chagas and dengue disease spreading Slimi et al. [2009] Pereira and Schimit [2018], traffic flow modeling Nagel and Schreckenberg [1992] and LUCC models Ghosh et al. [2017].

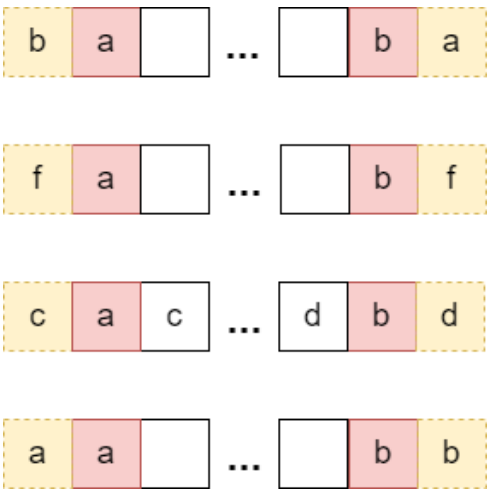
The difference between DCA and PCA can be illustrated by a classic epidemiological CA model SIR (Susceptible-Infected-Removed). Considering DCA, the transitional rules would give a single output state out of the three, in other words, based on the neighbors' states, the algorithm would simply search for the compatible rule and set the result. For a PCA, the state output would have a probability of occurrence, for instance, the more infected cells around the central cell, the higher the chances of the central being a future infected, but not strictly 100% Fu and Milne [2003].

2.1.4 Boundary conditions

The CA lattice of cells is essentially finite, once the model usually simulates a limited area of a phenomenon or event of interest. Obviously, the cells in the border have no neighbor cells available like the others. In this way, it requires defining special conditions in the boundary cells. One of the strategies for those cells is setting a particular set of rules, that considers the specifications of the boundaries. Another solution relies on extending the neighborhood in virtual cells in different ways. Figure 3 presents four typically used boundary behaviors that one can adopt, for illustration purposes using a 1D CA configuration. Virtual cells are represented with non-continuous lines and the lattice cells with continuous ones Chopard and Droz [1998].

The so-called periodic, or cyclic, considers that the borders are connected forming a torus-like topology. A fixed boundary forms a one-state ring cell that doesn't change over time. A reflecting boundary takes the state from the neighbor cell and into the virtual cell. The adiabatic way simply consists in duplicating the value of the boundary cell for the virtual one Kari [2005].

Figure 3 – Different types of boundary conditions for a CA modeling



2.2 Fuzzy Times Series

2.2.1 Time Series Concept

A collection of an orderly sequence of variable values is defined as Time Series. This special type of data requires appropriate techniques once the observations cannot be analyzed individually or randomly but considering the historical temporal dependence Wayne A. Woodward [2022]. Many types of phenomena and processes can be expressed as Time Series data, some examples are the collection of the temperature value at a particular location along the days, electricity consumption along the seasons, and solar irradiance over the hours. In this context, Time Series Forecasting (TSF) can be considered a significant tool in order to study and analyze those types of scenarios.

The forecasting method can be defined as a procedure for computing predict behaviors from present and past values. TSF models integrate pattern recognition techniques to create a system able to conjecture future scenarios based on past data and other external variables. Forecasting applications include observations of seasonality of electroencephalogram examination patterns Forney [2011], Chowdary et al. [2022], social media sentiment analysis Ibrahim and Wang [2019], deforestation prediction Dominguez et al. [2022], disease spread analysis Chyon et al. [2022].

As Chatfield [2000] describes, in general, there are three types of forecasting methods:

1. Judgemental forecasts based on specialist knowledge and human cognition. It is a subjective method since it requires human perspective, insights, and updated information about the scenario to be forecasted. In general, it is an option when there's a lack of historical data O'Connor et al. [1993];

2. Univariate methods where the scenario depends only on the historical data of one variable. Frequently augmented by a function of time such as a linear trend Moosa [2000];

3. Multivariate methods where forecasts of a given variable are based on observations of one or more additional time series variables. Multivariate forecasts may depend on a multivariate model involving more than one equation if the variables are jointly dependent.

A single forecast model can associate more than just one approach. For example, judgmental models are frequently involved in forecasting once technical knowledge is used to build or adjust the models and it can potentially improve the evaluation metrics such as accuracy Chatfield [2000].

2.2.2 Fuzzy Time Series Concept

Fuzzy Time Series (FTS) is a forecasting method, first proposed by Song and Chissom [1993]. The central principle is based on building a time series forecasting model through fuzzy set-based representation, in other words, instead of a conventional time series set, FTS represents a time series as fuzzy sets. As first defined in Zadeh [1965], a fuzzy set is a collection of objects, each of which has a continuum of membership grades. Membership functions assign a grade of membership ranging from zero to one to each object in such a set. Traditional concepts such as inclusion, union, intersection, complement, relation, convexity, etc., are extended to such sets.

FTS models are easy to implement and it allows high interpretability and flexibility, affording a variety of data types. It is a data-driven and non-parametric approach once it requires only a historical dataset and a non-fixed set of parameters that can vary accordingly to the model decisions. FTS models have been used in many areas of application, such as seasonal time series Song [1999], stock index prices Huarng and Yu [2005], electric load Efendi et al. [2015], and others Bose and Mali [2019].

As presented in Song and Chissom [1993], Fuzzy Time Series is defined as: Let $Y(t)$, $t \in \mathbb{Z}$, a conventional time series in a subset of $\mathbb{R}^1$. The universe of discourse is divided as $U = [min(Y), max(Y)]$ and the fuzzy sets A_i, $i \in \mathbb{Z}$ defined over the U intervals with corresponding membership functions f_{A_i}. Thus, considering $F(t)$ a collection of f_{A_i} extracted from original $Y(t)$, then can be defined as FTS on $Y(t)$.

FTS models $\mathcal{M}$ belong to a class of supervised learning method, once it requires a set of training and test data in order to build the model. Thus, the development is divided into two major processes: the Training procedure, where $\mathcal{M}$ is refined, and the Forecasting procedure, which computes the future values of the time series.

Model Training Procedure

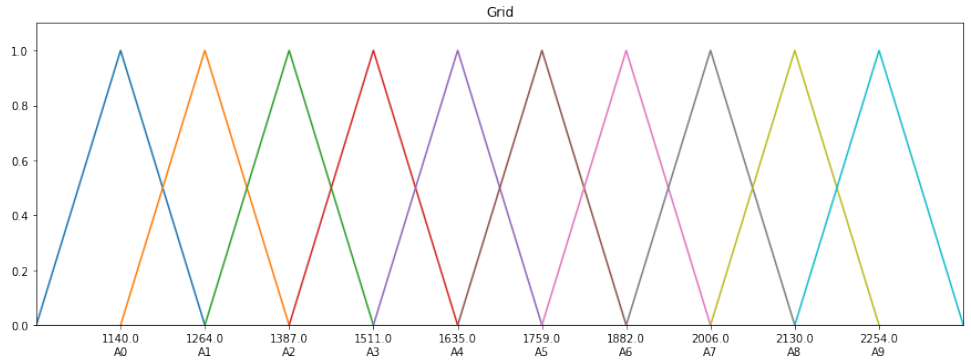
1. **Pre-processing:** Considered as an optional step, the training time series dataset $Y(t)$ can be pre-processed and some transformations can be applied. Frequently used transformations are Box-Cox, Differential, and Normalization, among others. Usually, the main goal is noise reduction, the universe of discourse range contraction, detrending, etc. In order to exemplify the next steps, consider the time series shown in Figure 4, presenting an original dataset and one possible pre-processing application:

2. **Universe of Discourse Partitioning:** The Universe of Discourse U, as described in the definition, is the closed interval with the minimum and maximum value of the time series $Y(t)$ as limit points. This interval is divided into k fuzzy sets A_i. Figure 5 presents a possible partition, considering triangular membership functions:

Where $\tilde{A} = \{A0, A1, A2.., A9\}$ is the set of fuzzy linguistics variables. And the membership functions $mf_{A_i} : \mathbb{R} \rightarrow [0, 1]$. As a traditional fuzzy system, membership

Figure 4 – Example of a Time Serie, on top the original data and below a transformation applied.

Figure 5 – Example of a Universe of Discourse Partitioning with ten fuzzy sets: from A0 to A9.

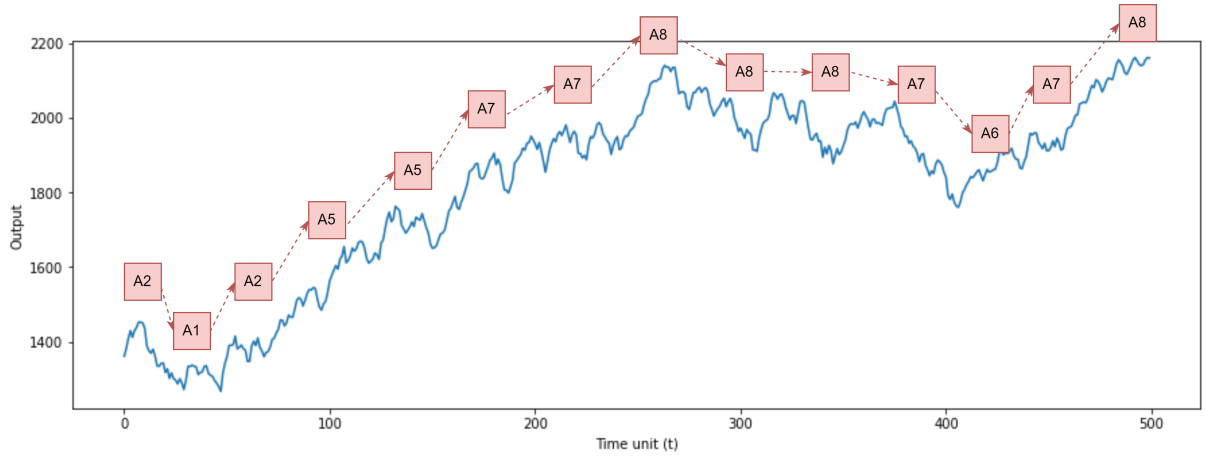


functions can be modeled in a variety of ways, the most common cases are Triangular, Sigmoid, Trapezoidal, and Gaussian de Lima e Silva [2019].

3. **Time Series Fuzzification:** In this step, the main idea is converting the crisp values of the conventional time series $Y(t)$ into the fuzzy representation composing a fuzzified time series $F(t)$, based on the previous step configuration. In other words, each value in $F(t)$ corresponds to a fuzzy set A_i that was taken from $Y(t)$. Figure 6 shows the evolution of the fuzzy set considering the partitioning in Figure 5 for the Time Series presented in Figure 4.

4. **Temporal Patterns Extraction:** Given a fuzzified time series $F(t)$, the Fuzzy Logic Relationship (FLR) set can be extracted accordingly to the number of past observations chosen for the model, defining the FTS order. For the first-order FTS model, the FLR establishes a fuzzy relationship (R) between $F(t)$ and $F(t-1)$:

Figure 6 – Fuzzified Time Serie



$$F(t) = F(t-1) \circ R(t-1, t) \tag{2.1}$$

Or, equivalently:

$$F(t-1) \rightarrow F(t) \tag{2.2}$$

Where $F(t)$ and $F(t-1)$ are the current and the first past observations. In general, an order-p FLR, determines the future observation based not only on the first past observation but the p lasts and is denoted by:

$$F(t-1), F(t-2), ..., F(t-p) \rightarrow F(t) \tag{2.3}$$

FTS models $\mathcal{M}$ with $p \geq 2$ are known as High Order FTS. Equation (2.3) shows that the weight of each $F(t-p)$ for obtaining the fuzzy forecast at time t, i.e. $F(t)$, is equal to one. Additionally, it can also be represented by fuzzy sets: $A_i \rightarrow A_j$, where A_i represents the left-hand side (LHS) or precedents and A_j the right-hand side (RHS) or the consequent of the FLR.

$$LHS \rightarrow RHS \tag{2.4}$$

The precedents are then grouped with their respective possible consequent, composing a set of Fuzzy Logical Relationship Group (FLRG). In order to illustrate how FLRG is composed, consider the following chronological occurrence of a fuzzified first-order time series, shown in Equation 2.5 below:

$$\begin{aligned} A_1 &\rightarrow A_1, \\ A_1 &\rightarrow A_2, \\ A_2 &\rightarrow A_2, \\ A_2 &\rightarrow A_1, \\ A_1 &\rightarrow A_3 \end{aligned} \tag{2.5}$$

Grouping equals LHS, the FLRG is established as:

$$\begin{aligned} A_1 &\rightarrow A_1, A_2, A_3 \\ A_2 &\rightarrow A_1, A_2 \end{aligned} \tag{2.6}$$

Note that, even though the FLR $A_1 \rightarrow A_1$ has more than one occurrence, under the FLRG perspective for a conventional FTS model, it has the same relevance as FLR that happens a single time. The interpretation of Equation 2.6 may be described as looking into the first past observation $F(t-1) = A_1$ of a variable from its historical dataset, composing the LHS_1 set, the possible next time-step observation is the set $\text{RHS}_1 = \{A_1, A_2, A_3\}$. A rule-based description is also possible:

IF "A_1" THEN "A_1 or A_2 or A_3 "

Forecasting Procedure

Once model $\mathcal{M}$ is trained, the forecasting method is used in the test dataset with the purpose to validate it based on selected evaluation metrics. The $\mathcal{M}$-input is the value from the original time series $Y(t)$ and the output is the future value $Y(t+1)$.

1. Input: Similar to the Time Series Fuzzification process from the training procedure. For each set of input values from the original time series Y, a fuzzy set is associated with it from F. The fuzzy set chosen is the one with the maximum image of the membership function for the element in the domain set, $Y(t)$:

$$F(t) = A_j \mid \mu_{A_j}(Y(t)) = max\{\mu_{A_1}(Y(t)), ..., \mu_{A_j}(Y(t))\} \tag{2.7}$$

2. Inference: The fuzzified precedents values are now searched in the set of FLRG rules from $\mathcal{M}$. A well-trained model is capable to hold a set of rules that describes the behavior of the time series studied scenario and provides a fuzzified consequent output.

3. Output: The output value is taken from the defuzzification concept of fuzzy logic. The fuzzified information obtained from the FLRG output $F(t+1)$ is now converted int to a crisp numeric value $\tilde{Y}(t+1)$.

4. Evaluation: The forecasting procedure creates a set of predicted values $\tilde{Y}$, that can be compared with the test-dataset through model metrics. This quantitative perspective shows the capacity of the model to simulate and predict the scenario.

2.2.3 Multivariate FTS

In terms of the number of input variables, a FTS model may vary accordingly to the studied scenario. Monovariate FTS has a single variable of interest and Multivariate FTS are models with more than just one. In general, the variables are divided into two types: endogenous, those influenced by other variables and associated with the target output variable, and the exogenous variable which works as extra and relevant information that helps to build the model. For a multivariate FTS model with dimensionality d and order p, is expressed by:

$$\begin{aligned} &(F_1(t-p), F_2(t-p), \ldots, F_d(t-p)), \\ &\qquad\qquad\qquad\qquad\qquad\qquad \vdots \\ &(F_1(t-2), F_2(t-2), \ldots, F_d(t-2)), \\ &(F_1(t-1), F_2(t-1), \ldots, F_d(t-1)) \rightarrow F_1(t), F_2(t), ..., F_d(t) \end{aligned} \tag{2.8}$$

Most of the real-world scenarios deal with multivariate models instead of monovariate, and the quantity can vary in terms of relation to the dynamical system. For example, in deforestation forecasting, the amplitude, the wind speed, the weather, and other parameters are pertinent in order to predict forest cover in future states Dominguez et al. [2022]. In this sense, an approach for spatio-temporal modeling was presented in Severiano et al. [2021], named Evolving Multivariate FTS (e-MVFTS), with an adaptation mechanism capable to deal with data distribution changes.

2.2.4 Weighted FTS

When extracting the Temporal Patterns from the fuzzified Time Series and creating the set of FLRG, some relevant information may be lost such as the frequency of a given rule and chronological order, as already mentioned and shown in Equation 2.6. The same rule may happen multiple times, in a given scenario, and it will have the same importance as a rule that happens just a few. Moreover, a pattern that happened in an older period has the same relevance as a recent pattern, losing the component of time, in a time-based model.

In this sense, Yu [2005] proposed the FTS method which considers the temporal pattern and the amount of time it appears in the fuzzified series using weights, known as the Weighted Fuzzy Time Series model (WFTS). For the Weighted Fuzzy Logical

Relationship Group (WFLRG) a weight matrix is included on FLRGs, giving greater importance to recent data in the forecast Silva et al. [2020].

In WFRLG, a fuzzy set repetition is considered as part of the grouping. Therefore, from the FLRG example shown in Equation 2.6, considering as WFLRG it becomes:

$$A_1 \rightarrow A_1, A_2, A_1, A_3 \tag{2.9}$$

The defuzzification processes take into account the midpoints of the WFLRG fuzzy sets. In general, considering a forecast of $F(t) = A_{j1}, A_{j2}, ..., A_{jk}$, the matrix of midpoints can be written:

$$M(t) = [m_{j1}, m_{j2}, ..., m_{jk}] \tag{2.10}$$

Different values of weights are now applied into the forecast $A_{j1}, A_{j2}, ..., A_{jk}$, satisfying the same condition as standard FTS models:

$$\sum_{h=1}^{k} w'_h = 1 \tag{2.11}$$

In this sense, the weight matrix is obtained following:

$$W(t) = [w'_1, w'_2, ..., w'_k] = \left[\frac{w_1}{\sum_{h=1}^{k} w_h}, \frac{w_2}{\sum_{h=1}^{k} w_h}, ..., \frac{w_k}{\sum_{h=1}^{k} w_h} \right] \tag{2.12}$$

where w_h is the weight for the A_{jk}. A suggested weights value described in Yu [2005], is setting the most recent forecast with the highest value and decreasing it until the older one. Equation 2.12 becomes:

$$W(t) = \left[\frac{1}{\sum_{h=1}^{k} h}, \frac{2}{\sum_{h=1}^{k} h}, ..., \frac{k}{\sum_{h=1}^{k} h} \right] \tag{2.13}$$

Thus, using the standardized $W(t)$ the final forecast value $Y_{(t+1)}$ is equal to the product of the defuzzified matrix and the transpose of the weight matrix Yu [2005]:

$$Y(t+1) = M(t) \times W(t)^T \tag{2.14}$$

where $M(t)$ is the defuzzified matrix forecast of $F(t)$ and $\times$ is the matrix product operator.

Since WFTS technique was presented, some improvements were developed, such as Trend WFTS Cheng et al. [2006], Improved WFTS Ismail and Efendi [2011], Exponentially WFTS Che Ngoc et al. [2018] and and Probabilistic WFTS de Lima Silva et al. [2020].

2.3 Related work

Cellular automata (CA) models have been broadly applied in spatio-temporal systems. Its simplicity, flexibility, and ability to incorporate the spatial and temporal dimensions of the dynamical scenarios are some of the strengths of these models and it allows the simulation of many types of real-world urban processes Santé et al. [2010]. As described in Aburas et al. [2016], the CA modeling strategy is a valuable technique in urban growth simulation and forecasting and shows that the integration with other techniques is a strategy to overcome some limitations. Although the flexibility is positive once it allows a widespread application, is a point of attention when trying to define a general CA model, remaining strictly to the specific situation. Other limitations involve the high dependency of programmers and researchers in order to build the model in the whole process: from transitional rules conceptualization to simulation's algorithms writing and the problem of being too simple and excluding from the model aspects as socio-economic factors Santé et al. [2010],Kari [2005].

In CA modeling, there can be found in several scientific fields and applications, such as ecological models of succession in vegetation Balzter et al. [1998], deforestation and fire propagation Zheng et al. [2017], Czerniak et al. [2018], urban growth Liu and Phinn [2003a], Mantelas et al. [2012], crowd evacuation Li et al. [2019], fluid dynamics simulation and physical systems Chopard and Droz [1998], Chopard et al. [2002], urban traffic simulation Tavares [2010], scattering study epidemics Melotti [2009], Áurea Tonetti Massahud [2011].

As mentioned in Chapter 1, the definition of a set of transitional rules may need extensive studies in order to achieve a good model able to simulate the real-world scenario. In this sense, transition rules based on artificial intelligence have been developed as an alternative to finding complex functions. Some examples are Support Vector Machines (SVM) Yang et al. [2008], least squares SVM Feng et al. [2016], Decision tree Li and Yeh [2004], Artificial Neural Networks (CA-ANNs) Yeh and Li [2003], Qiang and Lam [2015], k-NN algorithm Li and Liu [2006] usages, most of them associated with optimization algorithm and low interpretable methods. The CA-ANN integration strategy is able to build a CA simulation with good evaluation metrics, but with lower interpretability, since it is a Black-box approach and parameters must be set and optimized, such as the number of neurons in input, hidden, and output layers. Further, it has encountered problems dealing with time series images, for instance, which are extremely common in LUCC applications Wang et al. [2022]. In Li and Yeh [2004], a decision tree (See5 system) was used to generate the CA transitional rules, obtaining an overall accuracy of 75%. The strategy needs to deal with the optimization of the gain function and the vagueness of real-world scenarios was not considered with the Fuzzy Logic scheme.

As cited above, some techniques are combined with CA methodology, either to achieve better forecasting results by increasing evaluation metric results or searching for

better model parameters via optimization or calibration algorithms. In this sense, Cattaneo et al. [1997] first proposed a Fuzzy logic integration into CA models, known by as Fuzzy Cellular Automata (FCA). The strategy allows greater flexibility in the consideration of factors in the transition rules of states and modeling real-world scenarios' vagueness. Some usage examples are: a fuzzy-constrained CA model used to simulate forest insect infestations Bone et al. [2006], and logistic trends of urban development process study Liu and Phinn [2003b].

Fuzzy methods are often applied in order to simulate states gradient, where states are based on fuzzification of sharp and hard values by means of membership functions $mf_{A_i} : \mathbb{R} \rightarrow [0, 1]$. A fuzzy-neighborhood method was proposed by Małecki [2017], where instead of using traditional CA neighborhood configuration, a function is used to evaluate the influence of the neighboring cells, modeling it as fuzzy sets Ghosh et al. [2017]. Although it can be found in the literature a variety of approaches that incorporate Fuzzy logic into CA models, none of them has used Fuzzy Time Series methodology as a forecast model.

Chapter 3

Methodology

At a high level, the proposed method is essentially based on the development of a multivariate FTS forecasting model originating from a historical dataset of a geospatial dynamical event. Then, the model is used in a cellular automaton simulation, working equivalently as classical local rules set. In that way, for each cell in a traditional CA lattice, the FTS model is applied to compose the complete future CA grid scenario. In this chapter, the methodological approach is detailed and divided into two stages: Training and Forecasting procedures.

In terms of programming language, simulations were conducted using Python. Besides its outstanding code readability and high accessibility, Python is powerful in artificial intelligence (AI) and machine learning applications due to the variety of libraries and frameworks, such as NumPy, Keras, TensorFlow, and Seaborn, very useful to develop AI algorithms Raschka [2015]. In addition to the mentioned libraries and another reason to select the chosen language, the FTS models were built with Fuzzy Time Series for Python (pyFTS) library de Lima e Silva [2018].

3.1 Training procedure

The starting point is considering a collection of time series historical spatio-temporal data Y, extracted from observations, sensors output, images, satellite-based maps, and so forth. In images and maps data-type, the CA lattice ($\mathfrak{L}_d$) is represented by the grid of pixels, i. e., a single cell is a pixel in an image and the pixel color intensity composes the state set $\mathcal{S}$, for instance. A group of sequential images is a historical CA grids dataset. In other types of observations, those CA concepts need to be reflected in the spatial dynamics time series data. In computing, this is classically done by considering a set of sequential matrices.

The training procedure is defined as the scheme to obtain the multivariate FTS model to be implanted in the CA simulation. As emphasized, each sample of the collection

Y is a frame over the unit of time, representing the geospatial states of the phenomena, i.e. the collection groups the data of each time unit. The training steps are described:

1) Data preprocessing: Depending on the study case, methods such as noise reduction, standardization, and normalization may be necessary. A classic data transformation is known as differentiation which changes the original universe of discourse U for a smaller and stationary space de Lima e Silva [2019]. Mathematically it is defined as $\Delta y(t) = y(t-1) - y(t)$. PyFTS library offers different types of pre and pos data transformation processing: Differential, Scale, Box-Cox, and Adaptive-Expectation.

2) Variables definition: The system variables are determined, as well as the variable of interest, and the values of their states are stored following the time-ordered data collection. Each variable represents the historical state pattern over the unit of time. The system variables rely on the CA neighborhood chosen for the model and the exogenous variables. For instance, considering Moore's neighborhood the variables would be all the cells surrounding the center cell. The exogenous variables are all the additional classes of information that directly or indirectly impact the dynamical behavior of the studied scenario. In deforestation simulations, for example, the weather, humidity, forest type, distance to water (river, lakes, sea, etc), and distance to the urban areas and roads, may be considered exogenous variables due to the potential of relevance to understanding the deforestation dynamics of an area.

3) Dataset split: As discussed in chapter 2, FTS methods are data-driven supervised models and it requires a training and testing dataset to build and validate the forecasting model. In that way, the prepared dataset is divided into those two databases. In AI applications, the division proportion (train:test) is bound to the complexity of the problem and the learning algorithm but usually $7:3$ or $6:4$.

4) pyFTS Model Build: The following sub-steps define the hyper-parameters selection for the FTS model such as Order and Lags, Partitioning type, Number of fuzzy sets per variable, Weighted vs Weightless methods, and Membership function.

4.1) Universe of Discourse Partitioning: The pyFTS library offers different types of partitioning in order to build the membership functions. For both of the cases analyzed in this work was Grid Partitioning which divides the universe into n overlapping equal-length intervals with triangular membership functions. Furthermore, all the variables have the same partitioning and number of fuzzy sets. This configuration is adjustable and several combinations can be tested in order to achieve better prediction results. Figure 7 represents an example of the variable North using this type of partitioning for 4 fuzzy sets: A1, A2, A3, and A4.

4.2) Data Fuzzification: The process by which numeric values are converted into fuzzy linguistic variables. The crisp value from the time series $Y(t)$ is now represented

by the maximum membership value fuzzy set: $F(t) = \arg\max_{A_i} \mu_{A_i}(Y(t))$, since the fuzzy sets are overlapped. The order and lag indexes are selected.

4.3) MVFTS Model training: In chapter 2.2, the Eq. (2.8) was presented for a generic case of a multivariate FTS forecast model. Converting it to consider a CA perspective, the equation becomes:

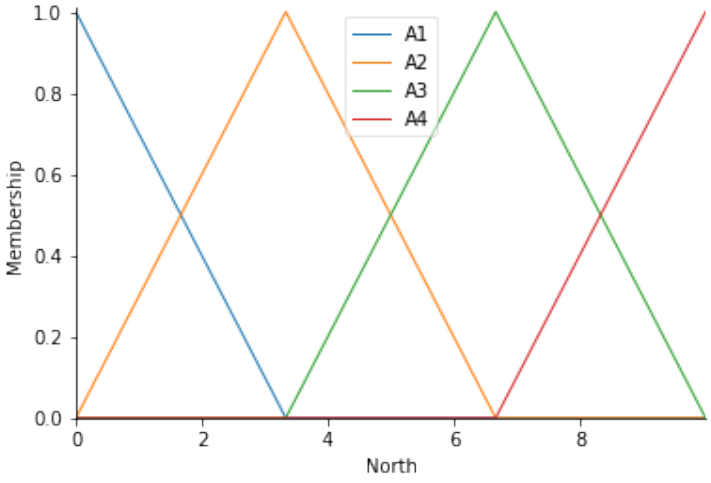
$$\begin{aligned} &(F_{NV}(t-p), F_{EV}(t-p)), \\ &\qquad\qquad\qquad\vdots \\ &(F_{NV}(t-2), F_{EV}(t-2)), \\ &(F_{NV}(t-1), F_{EV}(t-1)) \rightarrow F_C(t) \end{aligned} \tag{3.1}$$

Where p is the FTS model order, $F_{NV}(t-p)$ consist of the Neighborhood Variables (NV) and $F_{EV}(t-p)$ the Endogenous Variables (EV) of the CA-FTS system. The key idea considers the variables extracted from the CA lattice space and the additional relevant variable information. Thus, the temporal patterns and rules $LHS \rightarrow RHS$ are created. As example, considering Von Neumann's neighborhood, $F_{NV}(t-p)$ is written by:

$$\overrightarrow{F_{NV}(t-k)} = [F_N(t-k), F_S(t-k), F_W(t-k), F_E(t-k), F_C(t-k)] \tag{3.2}$$

Where N, S, W, E, C are North, South, West, East, and Center respectively for $\forall k = 1, 2, ..., p$. As a result of those steps, it is expected an MVFTS forecasting model, based on a set of historical spatial dynamical information, able to predict future state cells in CAs simulations.

Figure 7 – CA North Variable representation considering 4 fuzzy sets for a UoD = [0, 10]



3.2 Forecast procedure

The forecasting concept consists in incorporating the pyFTS model, trained from historical information, into a CA simulation. The algorithm, shown and detailed in 1, works by scanning the CA cells at time (t) and building it at time $(t+1)$, equivalently as the traditional state transition rules in CAs simulations. The fixed CA border considered, i. e., all the cells in the first or last column and in the first or last row, in a CA Grid, is set to a constant state and is not considered as a center cell, since there is no neighborhood. Finally, the predicted CA lattice is stored to be evaluated using selected metrics.

Algorithm 1 CA-FTS Forecasting

```
1: function CASimulation(CAGrid, ExogenousData, pyFTSmodel)
2:     while there're still cells on the CAGrid do
3:         currentCell ← CAGrid
4:         if currentCell is not at border then
5:             neighborhood ← CAGrid
6:             futureCell ← pyFTSmodel(currentCell, ExogenousData, neighborhood)
7:             predictedCAGrid ← futureCell
8: return predictedCAGrid
```

Figure 8 shows the summary of the steps through a diagram, with the training and forecasting procedure blocks.

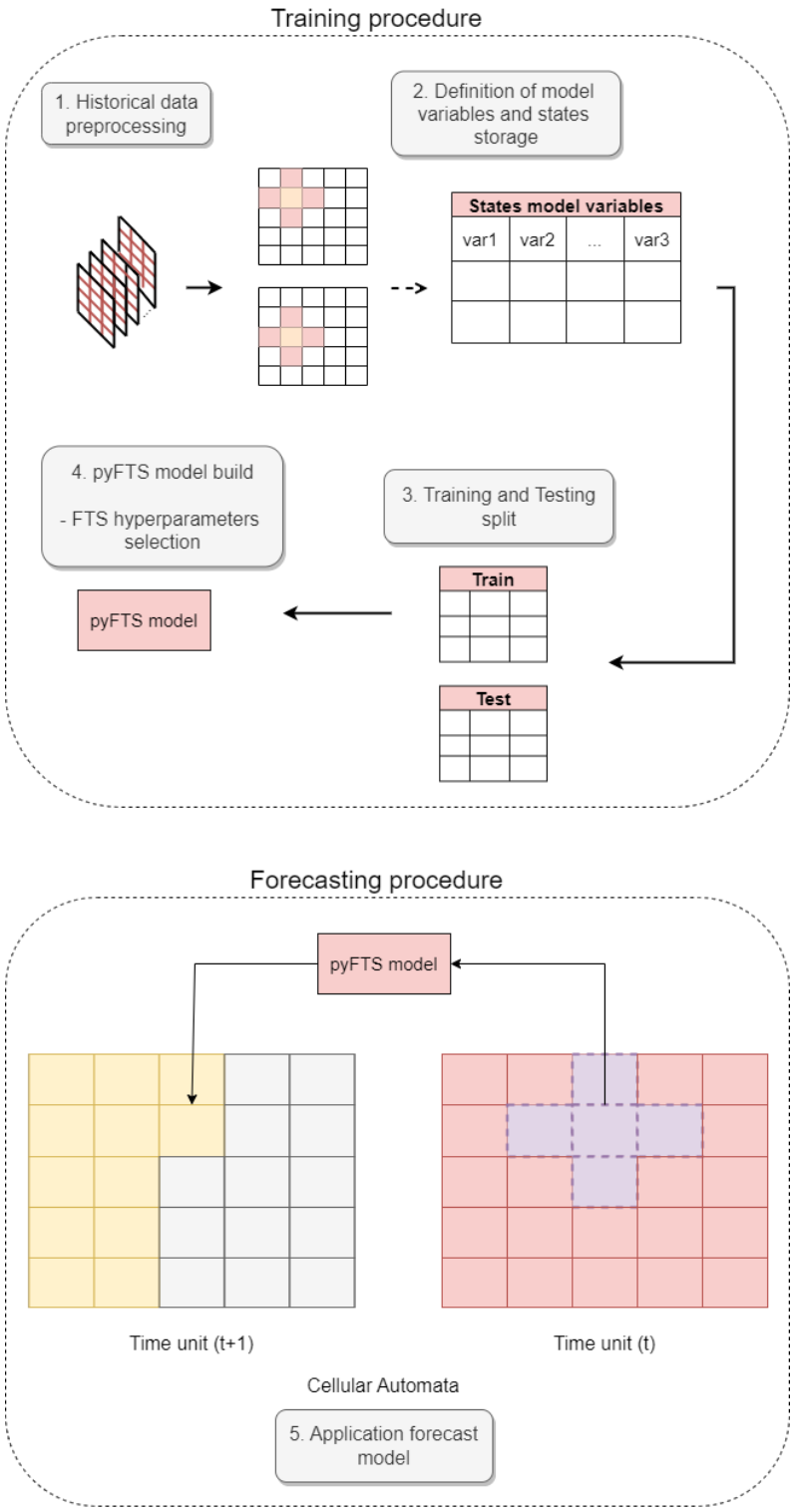
3.3 Evaluation Metrics

The evaluation metrics can be selected accordingly to the studied scenario. In this work, two options are suggested and used for two study cases. The first one is derived from a common measure of the quality of the model fit in AI systems and regression analyses, the Root Mean Square Error (RMSE), presented in Eq. (3.3).

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{3.3}$$

where n is the length of the test dataset, y_i is reference observation $\forall i = 1, 2, 3, ..., n$ and $\hat{y}_i$ the predicted value resulted from the pyFTS model. Larger values of RSME indicate a considerable difference between both observations. Therefore, a good model would produce lower values of the metric, $RSME \to 0$. The RMSE value is measured on the same scale and units as the original observations y_i, thus, normalization is performed on it in order to compare the model fits of different scales and ranges, for instance. The so-called Normalized RMSE is calculated as:

Figure 8 – Illustration of the proposed method steps.



$$nRMSE = \frac{\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}}{A} \tag{3.4}$$

considering A as the amplitude range of the data, considering both observed and predicted values. Another classic metric employed is known as Mean Absolute Error (MAE)

and is obtained by the sum of absolute errors divided by the size of the sample:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{3.5}$$

In terms of maps and images, the metrics are obtained from a cross-comparison between the predicted map produced, as opposed to a spatial reference. This reference can be extracted, for instance, from satellite imagery, on-site measurements using sensors, and social and aerial pictures from drones. In this context, the error matrix or transition matrix is a tool that allows calculating accuracy metrics for classification accuracy Diniz [2021]. Figure 9 presents the fundamental idea of the Transition Matrix.

Figure 9 – Generic Transition Matrix for cross-comparison metric

		Actual data			
	Classes	A	B	C	D
Predicted Data	A	TA	FP	FP	FP
	B	FN	TB	FP	FP
	C	FN	FN	TC	FP
	D	FN	FN	FN	TD

Source: Diniz [2021]

Classes A, B C, and D are features of a map. The columns represent the reference perspective and the rows are the predicted or classified map produced from the model. The green color, located in the main diagonal, represents the accurate classified information or the true positives (True A - TA, True B - TB, True C - TC, and True D - TD). The False Positives (FP), in the top-right of the matrix, present the improperly classified "A" to an expected value of "not-A". FN stands for false negatives and refers to the omission of the classifier, i. e., a class classified as "non-A", is labeled to "A" by the model. There are four main metrics derived from a Transition Matrix: Producer's Accuracy and User's Accuracy:

1. Producer's Accuracy: the sample fractions of pixels of each class correctly assigned to their classes by the classifiers.

$$PA = \frac{V_n}{\sum_r n} \tag{3.6}$$

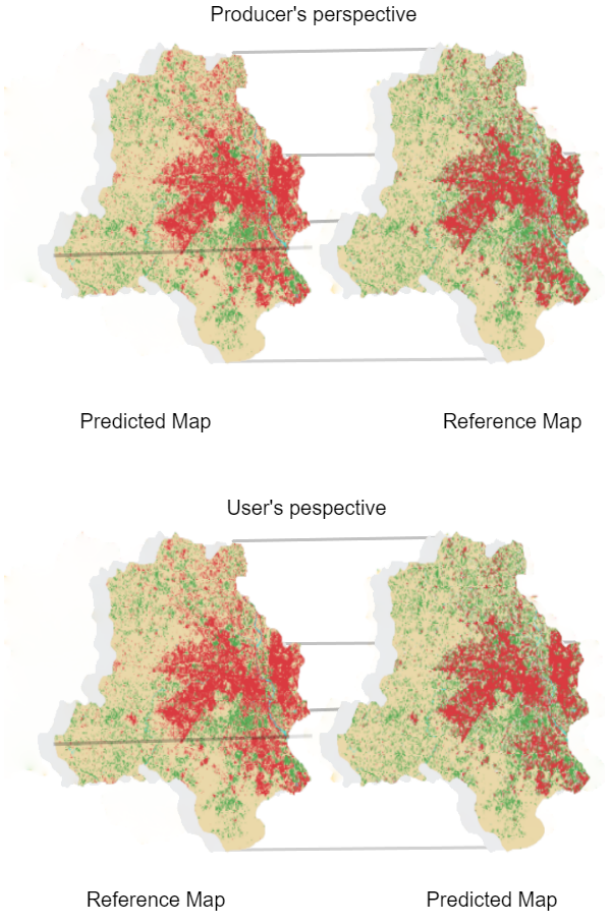
2. User's Accuracy: These are the estimates of the mapping pixel fractions, for each class, correctly classified.

$$UA = \frac{V_n}{\sum_c n} \tag{3.7}$$

where V is the amount of correctly classified pixels for class n and $\sum n$ the total amount of pixels.

Figure 10 presents a better visualization for both metrics using a Reference Map, extracted for example from satellite-based data, and a Predicted Map from an output of a simulation modeling. From a Producer's perspective, the Reference map is put in front of the Predicted Map and the pixels are compared, for the User's perspective the opposite is applied.

Figure 10 – Representation for Producer and User's Accuracy's.



3.4 Supporting tools

Map images and their formats were handled using the QGIS platform. Created by Gary Sherman in 2009, QGIS is a free-and-open-source (FOSS) application for geographical information system (GIS) data. It is possible to visualize, edit, analyze data, and compose maps. Hence, the studied classes in the predicted and actual maps, as raster layers format, were artificially colored by the software.

Chapter 4

Dataset Description

This chapter presents the background of the two datasets used to validate the Cellular Automata Fuzzy Time Series (CA-FTS) methodology. The datasets are spatio-temporal scenarios that primarily applied a typical CA model to simulate the states change dynamics. The first one is an epidemiological model for Chagas disease spreading simulation, from Slimi et al. [2009] and the second is a land use and cover change study in Delhi, India Tripathy and Kumar [2019]. The following sections intend to present the model and the methodology used by the authors and highlight the complexity when building transitional CA rules. It is relevant to mention and emphasize that the proposed methodology suits for any historical data from spatio-temporal models. In this work, it was used two CA-based models in order to obtain the original dataset and validate it through the selected evaluation metrics.

4.1 Spatial Spread of Chagas Disease

Popular in Latin American countries, Chagas disease, or American trypanosomiasis, is an infection caused by the protozoan parasite *Trypanosoma cruzi* (T. cruzi). It was first identified in 1909 by the Brazilian physician Carlos Chagas, and it remains a public health and social issue, especially in neglected areas but also in non-endemic areas Chagas [1909], Suárez et al. [2022]. Accordingly to the World Health Organization (WHO), the estimated number of infected people is about 6-7 million worldwide. The infection may cause cardiomyopathy, arrhythmia, and megaviscera in up to 30% of chronically infected people, and around 10% develop digestive or neurological problems Pérez-Molina and Molina [2018]. Treatments and drugs have been improved since its discovery and an early diagnosis can cure Chagas disease nowadays but no effective vaccine was developed until now. Thus, the prevention strategy remains in identified early infection, chemotherapy and vector control Bivona et al. [2020].

In general, T. cruzi parasites are transmitted by contact with the excretion of

an infected blood-sucking vector, the triatomine bug. As mentioned, one known effective prevention practice is the control of the vector agent and, as Slimi et al. [2009] pointed out, efforts have been made in order to properly understand the dynamics of vectors populations in order to align strategies and develop an appropriate mitigation plan Bartsch et al. [2017] Inaba and Sekine [2004]. In this sense, Slimi et al. [2009] presented a probabilistic cellular automaton model able to simulate the annual demography and dispersal of the vector during infection and non-infection periods in the Yucatan peninsula region, in Mexico. It is a spatio-temporal computational simulation of the population dynamics of the Chagas disease over time in a limited discrete space, that allows the study and understanding of the behavior of the bugs. The relevance and necessity of the study rely on the low efficiency of insecticides in some specific regions, where the occurrence of infected bugs is strongly bonded with seasonal immigration from forest to village, not in domiciled places, causing, for instance, re-infested dwellings Slimi et al. [2009].

The study gathers relevant information and real data from previous research to determine and calculate a set of parameters, variables, and probabilities used to build a classic CA-based model. It was considered a two-dimensional square lattice of cells $\mathfrak{L}$ of size $M \times M$. Each cell is composed of the number of larvae (n_y) and adult vector insets (n_a). The simulation considered a maximum limit of 5 for each stage of the bug life ($n_{amax} = n_{ymax} = 5$). The set of states $\mathcal{S}$ is based on the quantity of larvae and adults $\mathcal{S} = \{0, 1, ..., n_{amax}\} \times \{0, 1, ..., n_{ymax}\}$. A cell state is represented by the vector $[n_y, n_a]$ with the possibility of having characteristics of a garden (G) and house (H) habitat, although in the simulation was considered all the cells in H-habitat. The transition rules were built according to a typical CA, that is, bringing together specific knowledge and characteristics of the region. As described in the article, these rules were divided into Demography and Dispersion processes.

1. **Demography**: defines the number of individuals per cell and it is composed of the phases reproduction, survival, and development.

    a) **Reproduction**: the adults reproduces with probability p_r, generating $F = 1$ larvae. The number of larvae generated per cell is described as a random variable following a binomial distribution $B(F.n_a, p_r)$. The state of the cell after reproduction is:

    $$S_{t+1} = [n_y + n_{ry}, n_a] \tag{4.1}$$

    b) **Survival**: corresponding to the amount larvae and adults survived at each time-step. It is also given by a binomial distribution $B(n*, p*)$. The probabilities are p_{sy} and p_{sa} for larvae and adults respectively. The final state after the survival phase is:

    $$S_{t+1} = [n_{ry} + n_{sy}, n_{sa}] \tag{4.2}$$

c) **Development**: the processes of larvae becoming adult insects. Each larva has the probability p_d to develop and the binomial distribution takes the survival larvae $B(n_{sy}, p_d)$. After the development, the final states are:

$$S_{t+1} = [n_{sy} + n_{ry} - n_{da}, n_{sa} + n_{da}] \tag{4.3}$$

2. **Dispersion**: Defines the movement of the adult vectors over the cells in the lattice. A fixed boundary condition is considered and it relates to the migration of the insects from the forest to the village. During the infestation, the period is set to $Nf = 1$ bug and none during non-infestation. The probability or dispersal coefficient associated with the movement from the boundary forest to the village is:

$$D_f = \frac{Q_f}{4M} \tag{4.4}$$

where $Q_f = 50$ insects/day and M the lattice dimension.

The migration happens only in the infestation period (from April to June, or 90 days). Concerning the movement through the village cells in the lattice, each adult cells have the same probability to enter or leave the cell, given by:

$$p = \frac{D}{(2r-1)^2 - 1} \tag{4.5}$$

where D is the dispersal coefficient in the village, and r is the infestation radius (a higher value of r means more neighboring cells around the central one). In other words, for every time step, adults can leave the cell and move to a random neighboring cell with probability p from eq. (4.5), resulting a total reduction of n_{out} adults. Similarly, each adult can enter the cell from any neighboring cell with the same probability and the number of adults that arrives is n_{in}. The state of the cell after the Dispersion process is given by:

$$S_{t+1} = [n_y, n_a - n_{out} + n_{in}] \tag{4.6}$$

4.1.1 CA Set-up

As part of the implementation of a typical CA-based model, a set of parameters needs to be defined in order to achieve a model able to simulate the spatio-temporal behavior of the phenomena. In Slimi et al. [2009], the parameters were estimated based on real measurements of Chagas disease vector dynamics. In general, as mentioned in Chapter 2, this step usually takes time and effort, once involving for example, *in loco* data collection, years of research, and technical knowledge.

The estimation of survival and fecundity rates were investigated in houses from the Yucatan peninsula over the years 2000 and 2001. The developmental rate was calculated

by the mean time of development from egg to adult simulated on laboratory colonies of the bug from Yucatan. It was assumed that the survival, development, and fecundity are constant through time and the probabilities values for the time step were:

$$\begin{aligned} p_{sy} &= 0.90272518 \\ p_{sa} &= 0.9828095 \\ p_r &= 0.004111 \\ p_d &= 0.004158 \end{aligned} \tag{4.7}$$

The inside dispersion probability, presented in eq 4.5, the values of the coefficient D and infestation radius r are conditioned on the time of the year. During the infestation period, $D = 0.9$ and $r = 3$ and non-infestation $D = 0.1$ and $r = 1$. Let I be a subset of time-step t values in infestation period, in terms of probability p one can take:

$$\begin{cases} p = 0.5 & \text{if } t \in I \\ p = 0.4 & \text{if } t \notin I \end{cases} \tag{4.8}$$

Based on previous studies it was estimated that the range of values for the number of bugs that migrates from the forest to the village through the CA boundary is $10 \leq Q_f \leq 150$ bugs/day and a plausible one used in the model was $Q_f = 50$. The simulation was made considering a square CA lattice of cells sized 30×30, representing an area about $1km \times 1km$, i.e., each cell has $33 \times 33m^2$. The neighborhood considered was Moore neighborhood, shown in Figure 2 and the time-step t considered was one day, i. e., each new CA lattice composed from day t to $t + 1$ represents one day of a year.

It is relevant to emphasize that to achieve those specific transitional rules, technical knowledge was necessary to accurately reproduce the disease's spatio-temporal dynamics, with an elevated number of parameters. In this sense, Fraga et al. [2021] reproduced the Chagas disease CA model to propose a new strategy for adjusting transitional rules parameters by Genetic Algorithm (GA) with less error and better performance in the searching GA task.

4.1.2 Analysis

The model analysis used as the target variable the total sum of larvae (N_y) and adult individuals (N_a) in the village. The eq.(4.9) represents the sum in the lattice of cells $\mathfrak{L}$.

$$N_y^{(t)} = \sum_{c \in \mathfrak{L}} n_y^{(t)}(c) \quad \text{and} \quad N_a^{(t)} = \sum_{c \in \mathfrak{L}} n_a^{(t)}(c) \tag{4.9}$$

Where c is a cell, $n_y^{(t)}(c)$ is the number of larvae and $n_a^{(t)}(c)$ is the number of adults in c at the time step t. As described in the transition rules, after the demography process, the total number of adults individuals in a cell can be calculated with survived adults, the one developed from larva to adult and the total number of larvae in a cell is composed of that survived and haven't developed into adults and the news larvae from the reproduction of adults. These are represented with the equations:

$$N_y^{(t+1)} = p_{sy}(1 - p_d)N_y^t + p_r N_a^{(t)} \tag{4.10}$$

$$N_a^{(t+1)} = p_d p_{sy} N_y^t + p_s a N_a^{(t)} \tag{4.11}$$

After the dispersion process, the total number of larvae in the cell remains the same, once only adults are able to move from forest to village. The total number of adults is updated considering the amount of newly migrated adults (Q_f) during the infestation period and the adults that leave the lattice (Q_{out}) and defines the boundary condition. When the infestation period is over, $Q_f = 0$, i.e., no bugs enter the village, and the vector population tends to reduce.

$$N_a^{(t+1)} = p_d p_{sy} N_y^t + p_s a N_a^{(t)} + Q_f - Q_{out} \tag{4.12}$$

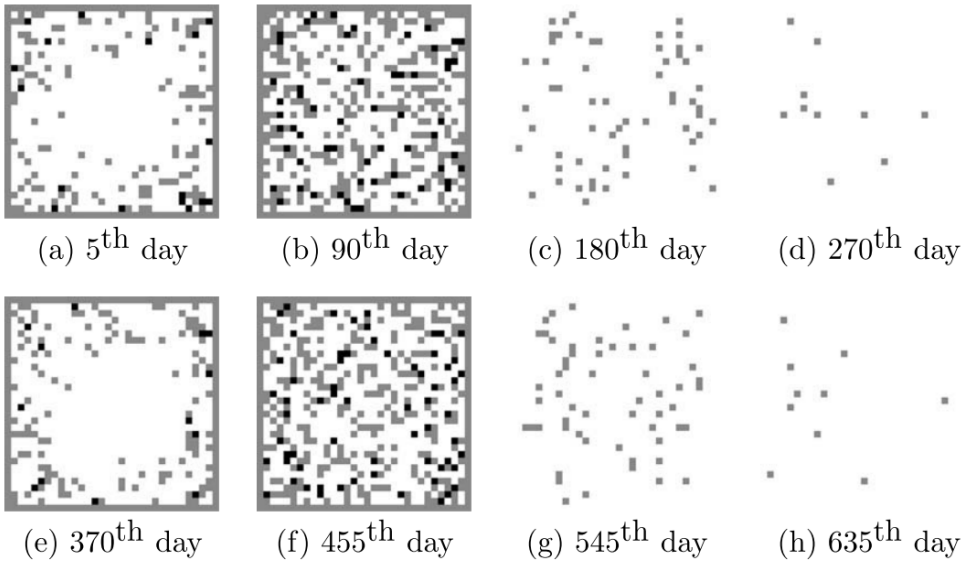
The mentioned analysis showed in eq. (4.9) was also used for the CA-FTS simulation to compare both of the results. As mentioned, the main intention of the proposed methodology is to build a model based on historical data, the analysis remains with the original strategy.

4.1.3 CA Simulation

Slimi et al. [2009] presented a visual representation of the spatial spread of the bugs over time, which helps to understand the behavior of the vector spread during infestation and non-infestation periods. Figure 11 shows the simulation, using C language, extracted from the original work. A shade of gray was used to represent the states associated with the number of adult individuals per cell: white means no adults, gray is when there is one adult, and black more the one adult. It is possible to see that during the infestation period, from day 0 to day 90, the boundary condition is fixed into $N_f = 1$ adult (gray color cells) and in the non-infestation is fixed to $N_f = 0$, from day 91 to 270 (white color cells). Furthermore, it can be noted that in the $90th$ day, the vector reached most of the village.

In terms of the model analysis described in the previous section, a graph was proposed to see the cyclic behavior over time. Figure 12 shows the simulation for 3700 days and the seasonal variations with a pick during the infestation period, where the total

Figure 11 – CA Simulation during infestation and non-infestation periods



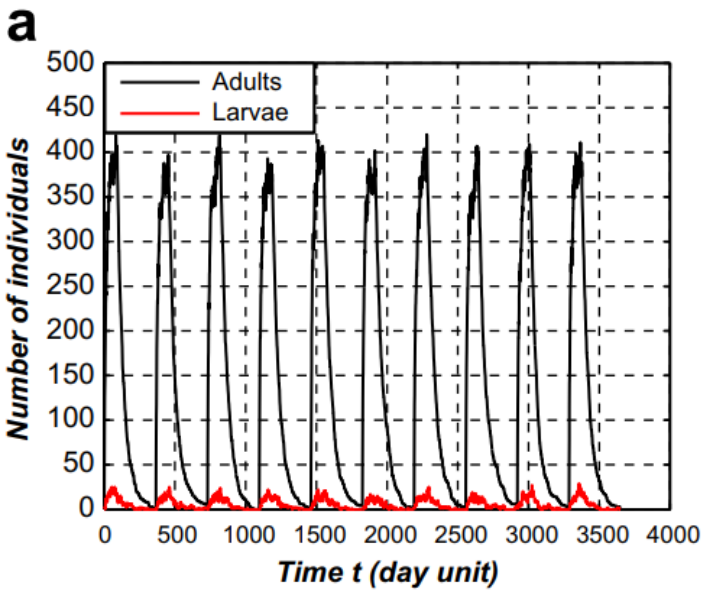
(a) 5^{th} day (b) 90^{th} day (c) 180^{th} day (d) 270^{th} day

(e) 370^{th} day (f) 455^{th} day (g) 545^{th} day (h) 635^{th} day

Source: Slimi et al. [2009]

sum of the bugs in the village is estimated at around 400, and during the non-infestation period, the amount reaches the zero. Both of those analyses were made by CA-FTS and it is presented in the next chapter.

Figure 12 – Evolution of the population of Adults and Larvae over time



Source: Slimi et al. [2009]

It is relevant to mention that the present work doesn't intend to deeply analyze the model developed in Slimi et al. [2009] in terms of Chagas disease spread dynamics itself, but demonstrates that it is possible to build an FTS model which can be used as transitional rules in a CA simulation to predict future cell states.

4.2 LUCC in Delhi, India

Urban sprawl modeling is a class of land use/cover change (LUCC) studies and has been widely applied to simulate and forecast future configurations on metropolises and urban centers Verburg et al. [2008]. It is often used as a tool to understand patterns and develop effective spatial policies Tripathy and Kumar [2019]. The process of building a model capable of reproducing spatial growth and decline of real-world urban areas is not a trivial task due to its complexity and several constraints inbred within the system Ghosh et al. [2017]. The relevance of this type of simulation, as mentioned in Chakraborty et al. [2022], is associated with the natural and human systems impacts of rapid, unplanned, and inefficient urban growth in different parts of the world. It has consequences in the natural environment, such as loss of biodiversity, and air or water quality which directly affect society, at local and global levels.

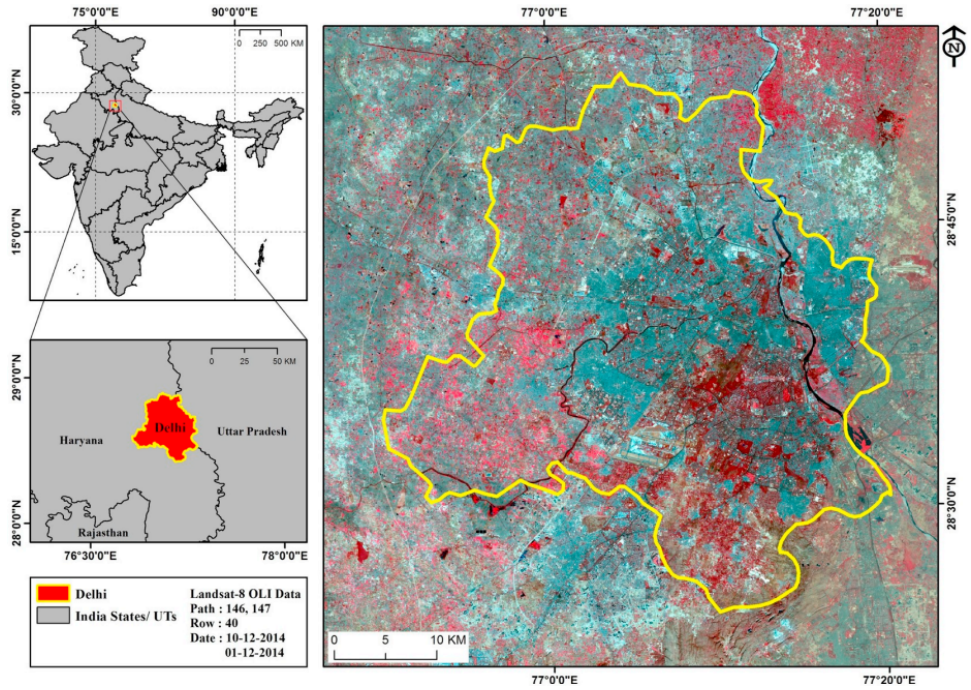
There are many different approaches and methods available in recent literature which propose paths to deal with various levels of complexity and build an urban growth model. One of the most used strategies, as exposed in Hassan and Elhassan [2020], are CA-based models. CAs models are often combined with other techniques, such as Markov Chain and Logistic regression, to identify specificities and achieve better results in simulating spatio-temporal LUCC dynamical scenarios. In this context, Tripathy and Kumar [2019] brings light to rapid urban growth in India causing socio-economic problems and large-scale impacts on ecosystems. Hence, it was developed a study case in Delhi, the capital of India which integrates geospatial techniques and CA modeling, using Geographic Information System (GIS) monitoring, to infer the urbanization process. The study gathers multi-temporal satellite imagery, Census of India data, and topographical sheets that helped to form thematic layers used as input variables to the CA system. In order to achieve the goal, it is executed a CA model calibration method to build the model and principal component analysis (PCA) technique to calculate the spatial variation analysis. It was considered four LUCC classes: Built-up, Vegetation, Water, and Others.

The capital of India, the city of Delhi, covers around 1483 km^2 of land and it is located in Northern India. Figure 13 presents the map of the study area. Delhi's population is projected to be over 19 million by 2021 Census [2011]. The population density of Delhi has increased from 9340 persons per km^2 in 2001 to 11320 per km^2 in 2011, the largest in India.

Available in the official USGS website[1], images of the LANDSAT satellite for the years 1989, 1994, 1999, 2004, 2009, and 2014 were extracted, composing a set of Land use/land cover maps. Note that a 5-year interval was taken into account between the annual map-based data gathered from the platform since the difference in the selected scale

[1] https://earthexplorer.usgs.gov/

Figure 13 – Satellite-based image of the study area - Delhi, India

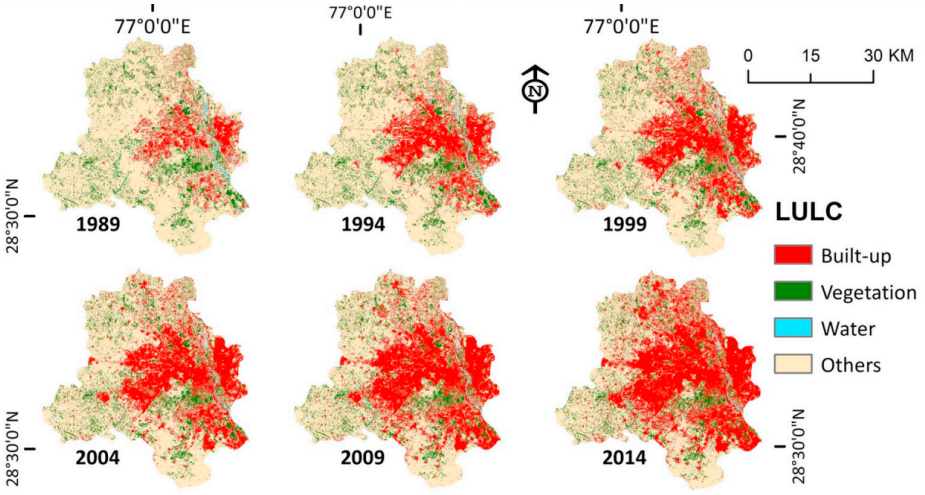


Source: Tripathy and Kumar [2019]

is more pronounced (and in terms of simulation, defines the CA time step). The maps were prepared as thematic layers, using the K-means classifier, in order to expose four classes of interest in the study. **(I) Built-up** areas are defined as man-made structures, such as residential and commercial areas, any type of buildings in urban and non-urban regions. It is characterized by high and moderate population density. **(II) Vegetation** gathers all green areas within the urban area and its surroundings, including agricultural land, parks, plantations, protected forests, and agriculture. **(III) Water**, as the name suggests, means all the surface waters, such as lakes, reservoirs, ponds, and rivers. The remaining area is classified as **(IV) Others** and composes every space except the three already mentioned classes. Examples of regions classified as Others could be undeveloped/fallow land, agricultural fallows, sandy areas, rocks, etc. Figure 14 presents the maps, with the four target variables highlighted by the colors: in red, the Built-up, in green, the vegetation, in blue for Water, and the others with light brown.

It was considered five supporting factors, also prepared as thematic layers, that directly impact the urbanization process: Population Density, Road network - Proximity to roads, Central Business District (CBD), Slope Map, and Restricted areas. Figure 15 shows the maps processed as thematic layers for the factors. The Population Density maps were based on the Census of India population data from the years 1991, 2001, and 2011. For 2019 and 2024, a projection was considered by the United Nations Population Projection 2030 Nations [2015]. The density maps were computed for different zones of Delhi, by vector layers. The slope map was based on Cartosat - I Digital Elevation Model

Figure 14 – Maps for the years 1989, 1994, 1999, 2004, 2009, 2014



Source: Tripathy and Kumar [2019]

(DEM) acquired from BHUVAN and classified into three degrees ($s < 3$, $3 < s < 5$, $s > 5$). The proximity to major roads map was generated by the major road network information from the topography sheet and classified at 500 m intervals up to 5 km, and 1000 m intervals beyond it. The Proximity to CBD was similarly defined, taking 5 km intervals in 10 levels. Lastly, the Restricted areas acquire from Delhi Development Authority Tripathy and Kumar [2019].

4.2.1 CA Set-up

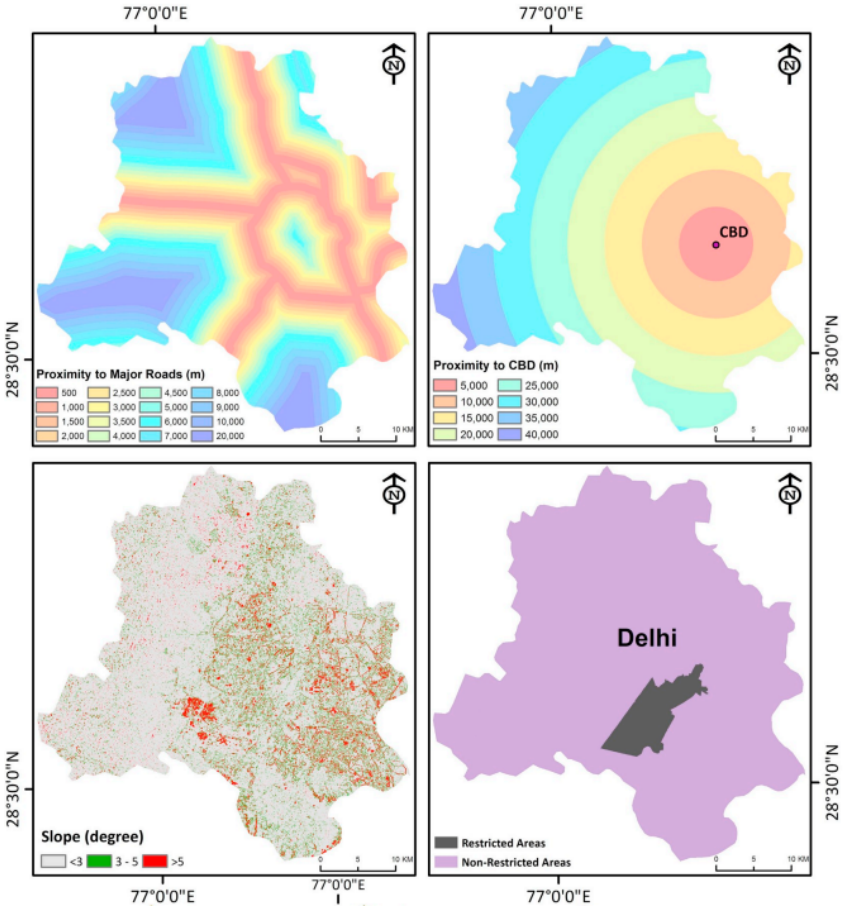
Firstly, for the simulation proposed, all of the maps' raster layers were set to 30×30 meters-sized cells, composing an equivalent geometry of the participating layers. It was considered the Moore neighborhood, defining a 3x3 size kernel:

$$A_{i,j}^{t} = \begin{bmatrix} a_{i-1,j-1}^{t} & a_{i-1,j}^{t} & a_{i-1,j+1}^{t} \\ a_{i,j-1}^{t} & a_{i,j}^{t} & a_{i,j+1}^{t} \\ a_{i+1,j-1}^{t} & a_{i+1,j}^{t} & a_{i+1,j+1}^{t} \end{bmatrix} \tag{4.13}$$

Where i, j specifies a single cell in a $d \times d$ sized CA lattice simulation. In terms of raster layers maps, $a_{i,j}^{t}$ represents a pixel in the image ($\forall \;\; i, j = 1, 2, .., d$) and its value determine a cell state at times (t). Raster layers must have the same size, once the current pixel (t) associates with the exact piece of land in the future state ($t + 1$).

As the CA definition states, the transition rules set ($\emptyset$) establish the relationship between the current cell state with the future one. The study Tripathy and Kumar [2019] considered a CA time-step to 5 years, for example for a given current year t_1, the future is ($t_1 + 5$) years. To define $\emptyset$, let T be the set of threshold values for the affecting parameters

Figure 15 – Exogenous variables (a) Proximity to major roads, (b) proximity to Central Business District (CBD), (c) Cartosar-I DEM-based slope map, (d) Restricted areas



Source: Tripathy and Kumar [2019]

and B the set of built-up count values in the kernel associated with every set of T. Ø was defined as functions ϕ of a set of conditional statements with threshold values:

$$\phi = f(T, B) \tag{4.14}$$

$$\begin{aligned} T &= \{T_R, T_C, T_P, T_S\} \\ B &= \{B_R, B_C, B_P, B_S\} \end{aligned} \tag{4.15}$$

where T_R, T_C, T_P and T_S are the threshold for proximity to roads, distance from CBD, population density, and slope value respectively. Similarly, B_R, B_C, B_P, and B_S are the corresponding number of built-up pixels in the test kernel for every element that belongs to T. The rules and specifications followed three main principles:

1. The classes for constructed land and bodies of water were spared from being eliminated, i. e. they remain constant.

2. A pixel can only be modified if, in the restricted areas layer map, it falls under the non-restricted category.

3. The Vegetation and Others classes may change into built-up land based on threshold values (T) and nearby built-up pixel counts (B).

Besides the definition of the transitional rules, the model was calibrated to find the best threshold values. The strategy consisted in taking the data from a year and simulating the future of the nearest year, e.g. the map for 1989 (t) was used to simulate the map for 1994 ($t+1$) and so on. Simulating the LUCC of time ($t+1$) using the LUCC of time (t) and the parameters of the supporting factors, the model was able to determine the best-set threshold values, by trial and error, that reflect the best possible results.

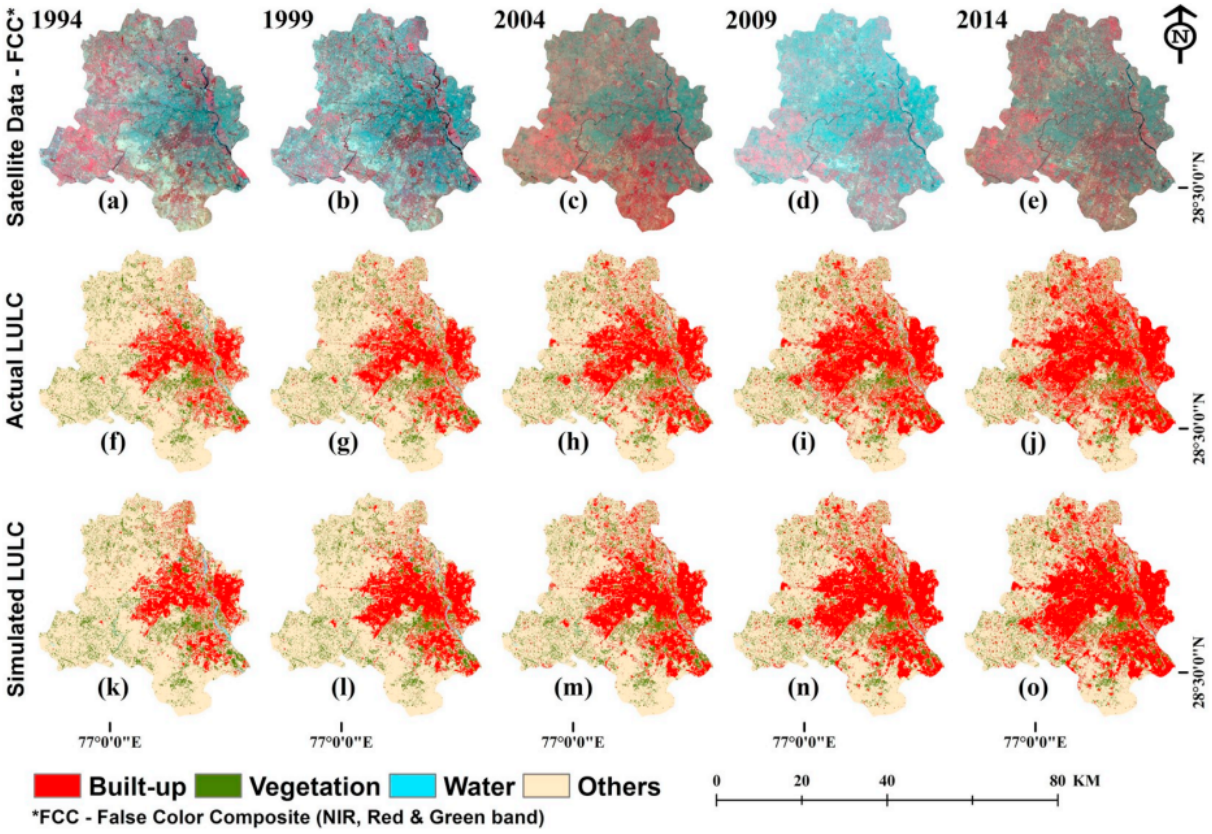
The difference between two images is calculated at each time step using the principal component analysis (PCA) technique, which works by spatially subtracting the built-up pixels of time ($t+1$) from the corresponding pixels of time (t).

4.2.2 CA Simulation and Discussion

The main idea behind the proposed methodology of Tripathy and Kumar [2019] starts with the set of Delhi maps acquire from satellite, used to observe the real changes in land use/land cover and urban growth patterns, between the years 1989 and 2014. Then, the calibration of the CA model was used to adjust the parameters in order to build the forecast model. Finally, the urban growth model is used to predict the dynamical behavior and compare it to the original maps through validation of simulated LUCC. Furthermore, the model was also used to predict the 4-classed maps for the years 2019 and 2024.

Figure 16 presents the satellite imagery in the first row, followed by the processed LUCC considering the four classes of interest, and in the last row, the results from the simulation. It is notable the similarities between the actual and simulated LUCC maps, indicate the ability of the model to catch the changes in the classes over time. Regarding the evaluation metrics, the principal component analysis (PCA) technique was applied indicating similarity classification in the average of 80% and the overall accuracy was 95.62% of the confusion matrix Tripathy and Kumar [2019].

Figure 16 – LANDSAT satellite images (a to e), satellite-based LUCC maps (f to j) and simulated LUCC map (k to o) of Delhi for the year 1994 (a, f, k), 1999 (b, g, l), 2004 (c, h, m), 2009 (d, i, n) and 2014 (e, j, o).



Source: Tripathy and Kumar [2019]

Chapter 5

Computer Experiments

In this chapter, the computer experiments of the proposed method are presented. For both of the study cases, the evaluation metrics are calculated and a discussion on the potential of the CA-FTS methodology is exposed. The Chagas disease CA model was reproduced in order to get the historical dataset and the Delhi datasets are available online in Amit Kumar [2019]. The cellular automaton based on fuzzy times series (CA-FTS) strategy is applied for both study cases.

5.1 Spatial Spread of Chagas Disease CA-FTS

Initially, the Chagas CA model was reproduced, considering the traditional rules described in the previous chapter, and simulated over 1730 days resulting in a set of historical data. A single day corresponds to a 2-dimensional matrix, representing the CA lattice, where each element is a CA cell composed of a number of adult vector individuals and larvae $[n_y, n_a]$ within it. Figure 17 illustrates through a simplified 7×7 CA Chagas Lattice with the number of individuals for each cell for a given random day N. The reproduction considered the same lattice size as the original work, a 30×30 grid where each cell represents $1km \times 1km$ in the real world. The total set of time series data is a collection of 1730 matrices containing the daily information of the total number of adults and larvae $N_a(t)$ and $N_y(t)$ from eq. (4.9). It is relevant to emphasize that this set is the only input data to build the proposed CA-FTS model, in other words, starting from the historical time series data, the CA-FTS model is automatically generated and used as transition rules to reconstruct the original data.

Once the dataset is built, the following steps consist of developing the proposed model. The variables of interest are defined accordingly with the chosen neighborhood and other possible exogenous variables. For the Chagas disease study case, there was no additional variable to the system besides the neighborhood and the central cell information. For each time-step, a day in the village, the grid was traversed in order to extract the

Figure 17 – Example of the simplified CA Lattice for a day

[1,3]	[1,1]	[0,0]	[2,2]	[0,0]	[2,2]	[0,0]
[1,2]	[2,2]	[0,0]	[2,2]	[1,1]	[0,0]	[2,2]
[1,1]	[1,3]	[1,1]	[2,2]	[0,0]	[0,0]	[1,1]
[2,2]	[0,0]	[1,2]	[0,0]	[1,2]	[0,0]	[2,2]
[1,2]	[1,1]	[1,0]	[0,0]	[1,2]	[1,2]	[0,1]
[5,1]	[2,0]	[1,1]	[0,1]	[0,0]	[1,1]	[5,5]
[3,1]	[1,2]	[1,3]	[0,0]	[4,1]	[1,2]	[0,0]

variables of interest from the model. A table was created where each row refers to the configuration of states at an instant (t) that conditioned the state of the central cell at $(t+1)$.

As discussed in Chapter 2, the FTS method is a supervised learning algorithm, thus the total dataset reproduced was split into train and test. The data from 2 years simulation, from day 0 to day 730, was separated for the training and the other part to test, from day 731 to 1730 days. The training data is used to build the FTS model from the pyFTS library, based on historical data of the variables of the constructed table. The Weighted-MVFTS method was selected due to its propriety of identity and gives more importance to the most frequent rules, instead of some that rarely happen during the dynamical state changes.

Finally, the model was applied to the CA simulation. The test dataset was used and for each time (t) in each cell of the lattice, the WMVFTS model determines the state of the cell at $(t+1)$. In other words, at this point, the method proposed works as a standard CA simulation, using as transitional rules the WMVFTS model built from the historical data of the phenomenon. All of the CAs lattices, or matrices, were stored to compare the results with the original dataset, extracting the evaluation metrics and the representation analysis. In terms of parameters used to evaluate the model, it was analyzed Moore's and Von Neumann's neighborhood, with radius 1, as the model's variables. For both situations, the WMFTS model was run with 5, 10, 20, and 30 fuzzy sets.

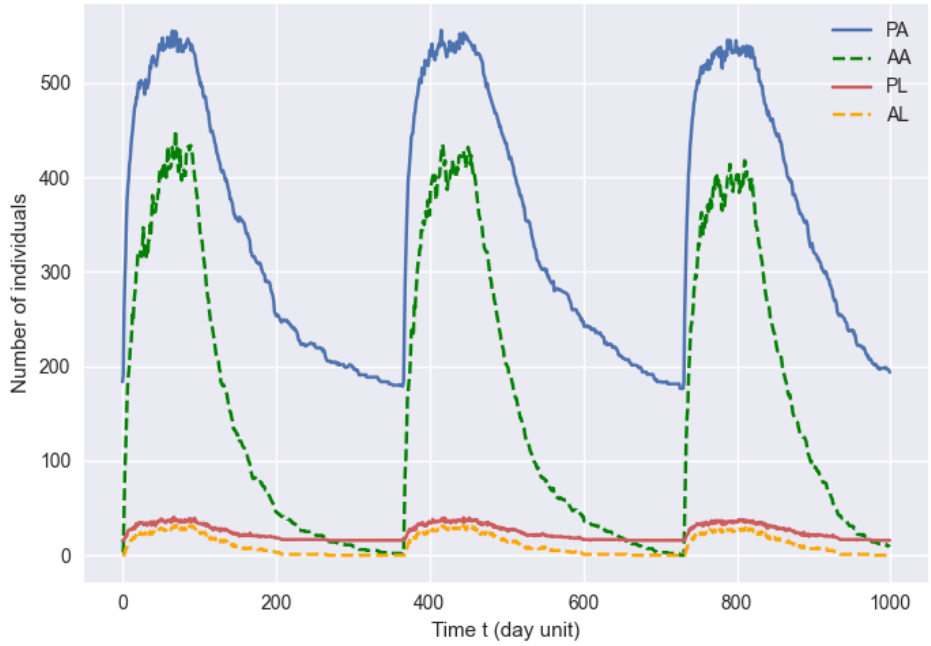
5.1.1 Results and CA-FTS Model Analysis

The graphs in Figures 18, 19, 20, 21 illustrates the simulation results. Green colored lines represent the test data for adults (AA), in blue the forecasting results using the CA-FTS model for adults (PA), in orange the test data for larvae (AL), and in

red the predicted values for larvae (PL). In other words, the continuous lines are the model-predicted values and the dashed lines are the actual values from the test dataset.

Figure 18 – Simulation results considering Moore Neighborhood (top) and Von Neumann (bottom) with 5 fuzzy sets.

(a) Moore



(b) Von Neumann

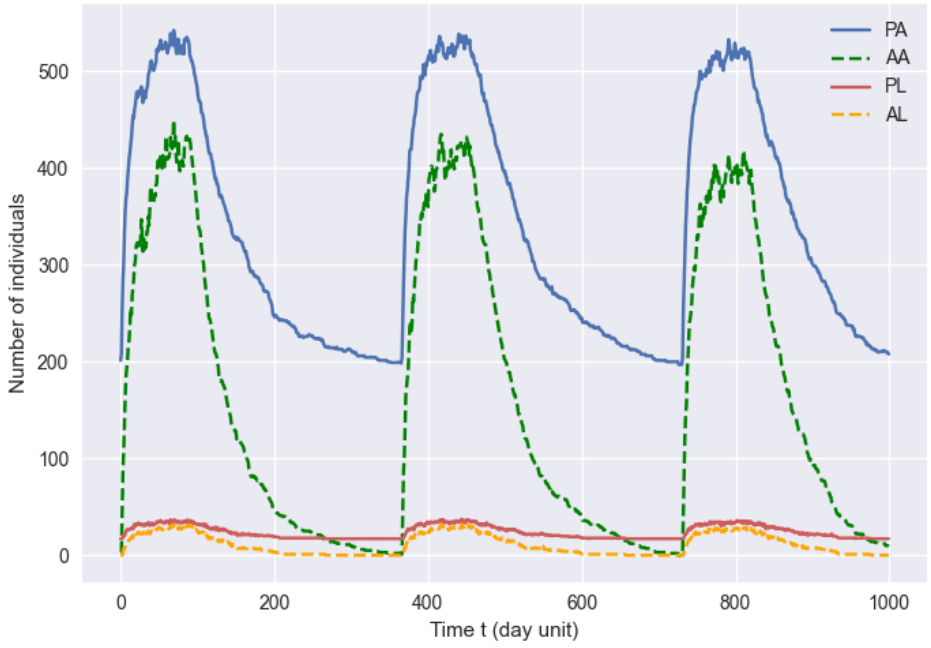
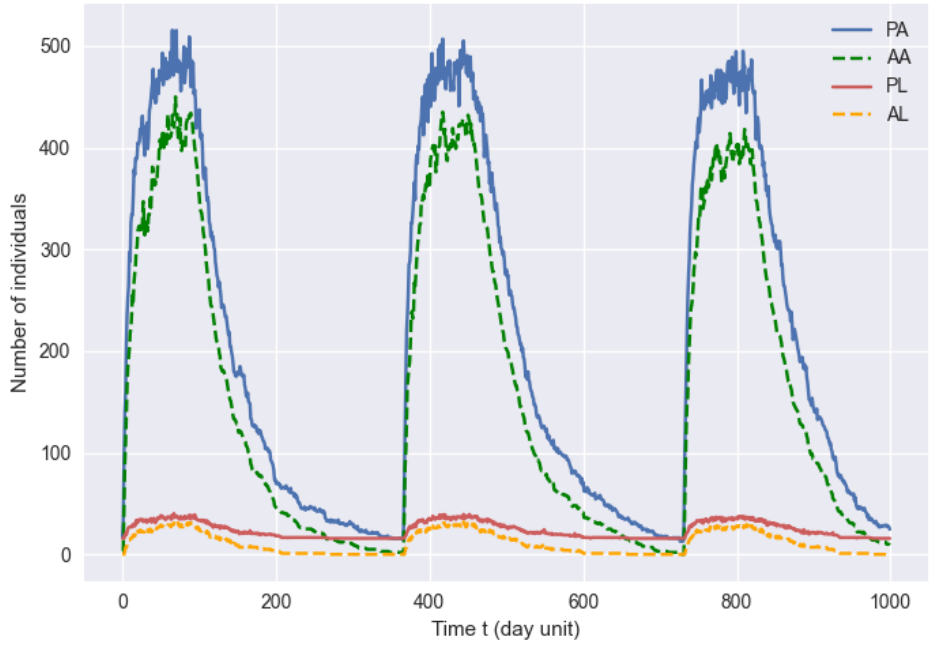


Figure 19 – Simulation results considering Moore Neighborhood (top) and Von Neumann (bottom) with 10 fuzzy sets.

(a) Moore



(b) Von Neumann

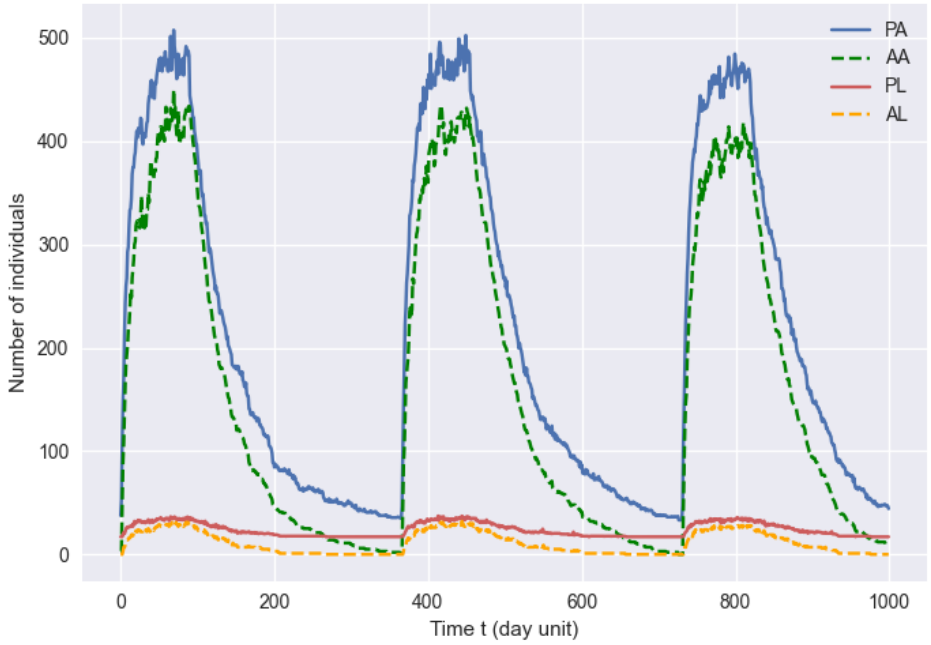
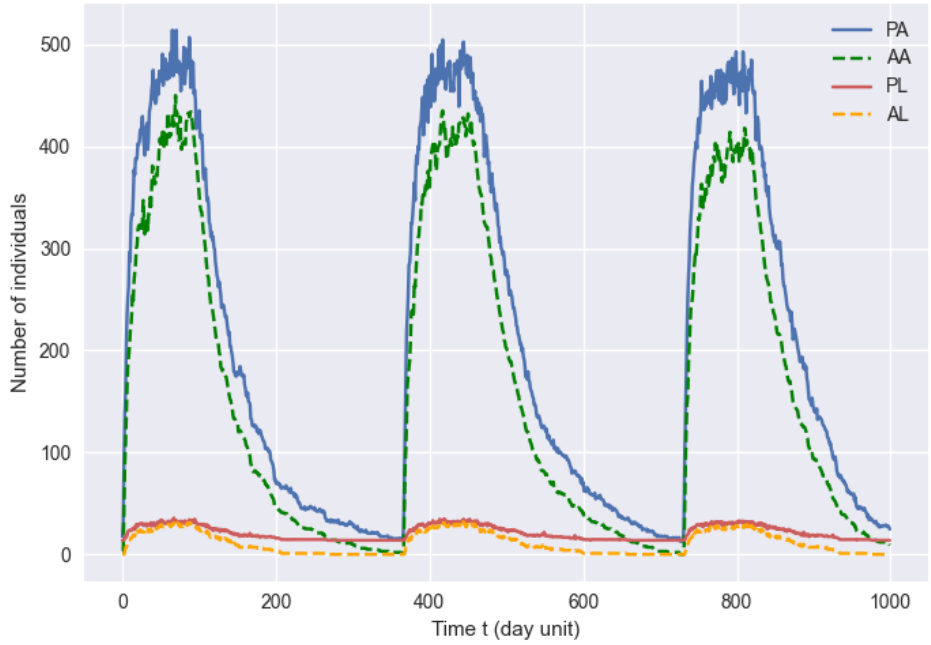


Figure 20 – Simulation results considering Moore Neighborhood (top) and Von Neumann (bottom) with 20 fuzzy sets.

(a) Moore



(b) Von Neumann

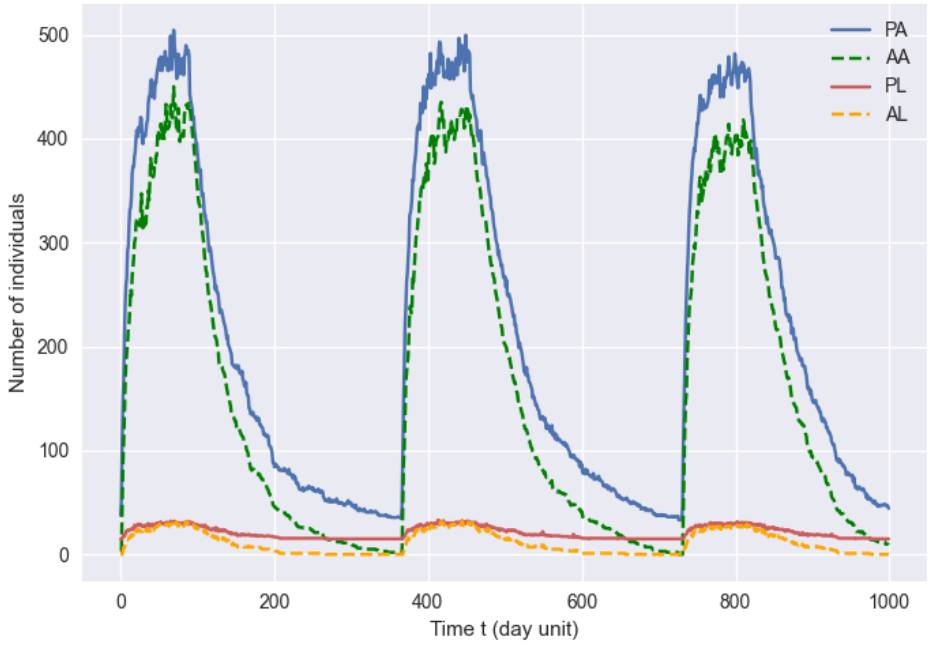
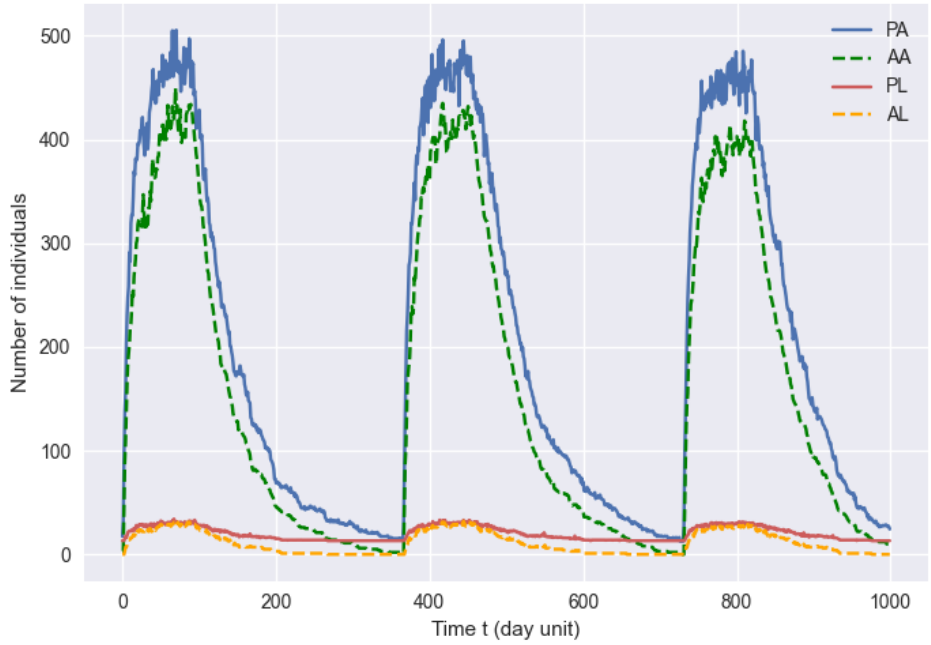
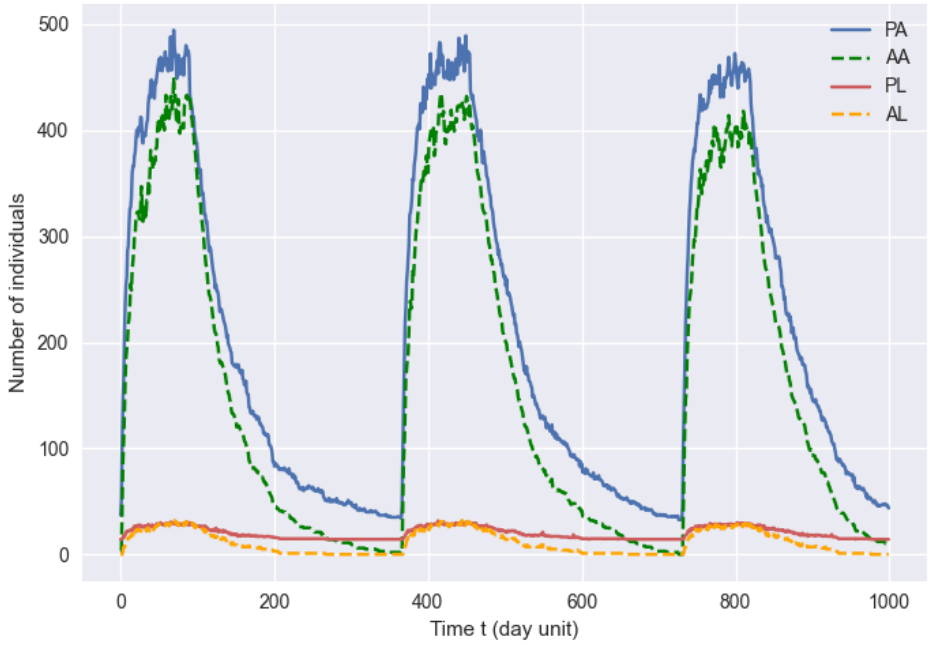


Figure 21 – Simulation results considering Moore Neighborhood (top) and Von Neumann (bottom) with 30 fuzzy sets.

(a) Moore



(b) Von Neumann



The metrics were obtained by comparing the original test data with the results from the CA-FTS model simulation. Tables 1 and 2 present the respective values of nRMSE and MAE according to the FTS model order and the number of fuzzy sets used, for adults and larvae respectively. The results were around 12% for nRSME and reached a Mean Absolute Error (MAE) ranging from $179,11$ to $8,96$. The best configuration obtained was with the FTS model using the Von Neumann neighborhood with 30 fuzzy sets, for adults and the Moore neighborhood with 30 fuzzy sets for larvae.

The simulation results, as observed in Figures 18, 19, 20, 21, demonstrate the model's capacity to capture and predict the expected pattern of the Chaga's infestation along the days, based exclusively on a set of phenomenon historical data. The graphs obtained are very similar to the original work, shown in Figure 12, where the peak, during the infestation period hits from 400 to 500 for adult individuals. According to the evaluation metrics, higher fuzzy set values resulted in better results for both configurations. This alternative strategy for CA modeling and simulation has a meager computational cost, around two minutes to build the model, and 1 hour to go over the prediction. Most of the processing time relies on the CA simulation itself. Furthermore, the functions used are available in the pyFTS library therefore the construction of the model is simple and easy to manipulate.

Table 1 – Evaluation metrics for Adults

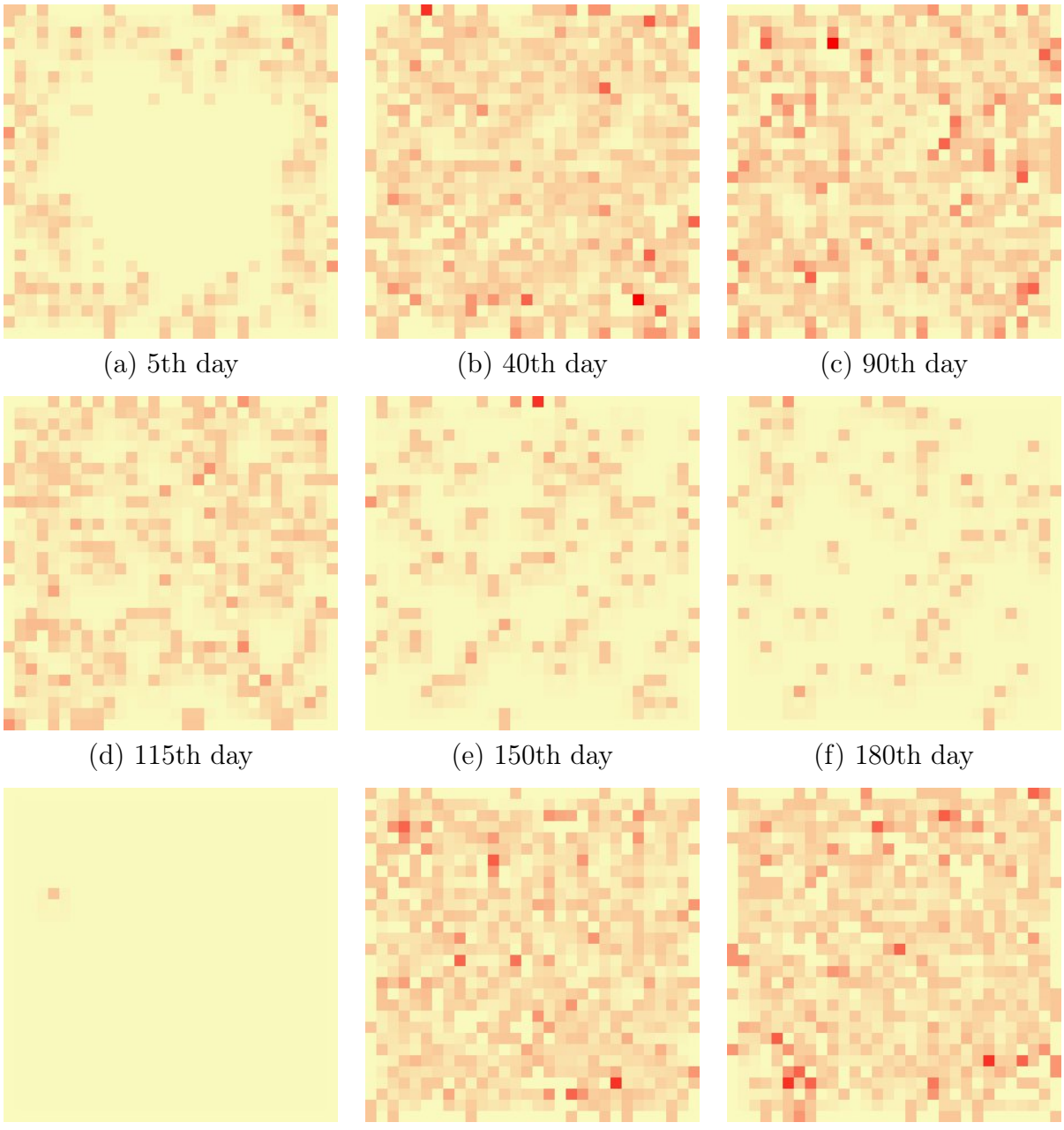
Neighborhood	N° of Fuzzy Sets	nRSME	MAE
Von Neumann	5	42,41%	179,11
	10	12,93%	53,07
	20	12,69%	52,14
	30	11,71%	48,13
Moore	5	44,83%	189,94
	10	13,01%	48,81
	20	12,79%	48,06
	30	11,81%	44,71

Table 2 – Evaluation metrics for Larvae

Neighborhood	N° of Fuzzy Sets	nRSME	MAE
Von Neumann	5	38,73%	12,91
	10	38,51%	12,80
	20	31,59%	9,88
	30	29,50%	8,97
Moore	5	38,04%	12,98
	10	37,79%	12,88
	20	30,53%	9,94
	30	28,31%	8,96

Following the same strategy as the original work Slimi et al. [2009], the representation of the dynamical spread of the vectors over the lattice is also applied. It was

Figure 22 – CA-FTS Simulation for Chagas disease spread at six stages of Infestation (I) and Non-Infestation (NI) periods: (a) 5th day - I (b) 40th day - I (c) 90th day - I (d) 115th day - NI (e) 150th day -NI (f) 180th day - NI (g) 366th day - NI (h) 401th day - I (i) 430th day - I



(a) 5th day (b) 40th day (c) 90th day

(d) 115th day (e) 150th day (f) 180th day

(g) 366th day (h) 401th day (i) 430th day

considered the best configuration obtained: using Von Neumann neighborhood with 30 fuzzy sets . The results is shown in Figure 22, extracting six stages of the infestation. The color gradient represents the amount of the adults individuals within a cell: the closer to red, the more vectors, i. e., the light yellow color represents no adults in the cell.

As described earlier in this section, the starting point of the test dataset is the beginning of the third year (731^{th} day), as reference to the total dataset reproduced. In this

way, the first CA lattice image in Figure 22, represents the 5^{th} day of vectors infestation in the third year and it is possible to see that the bugs are concentrated in the boarder of the area, once the migration was started. The next image shows the peak of the infestation, in the 90^{th} day. The vectors are now totally scattered in the lattice and in some areas with a high number of vectors. The infestation period ends and in both of following images,180^{th} day and 270^{th} day, is notable the reduction of the bugs over the grid. The results obtained is very closed to what was presented in the original CA work, shown in the previous chapter in Figure 11. The infestation and non-infestation Chagas disease behaviors were captured by the model.

As seen in the applied evaluation metrics, increasing the number of fuzzy sets resulted in the best FTS model. In terms of metrics, it can be noticed that using Von Neumann's neighborhood seems a better option, once the number of variables in the model is reduced compared to Moore's neighborhood.

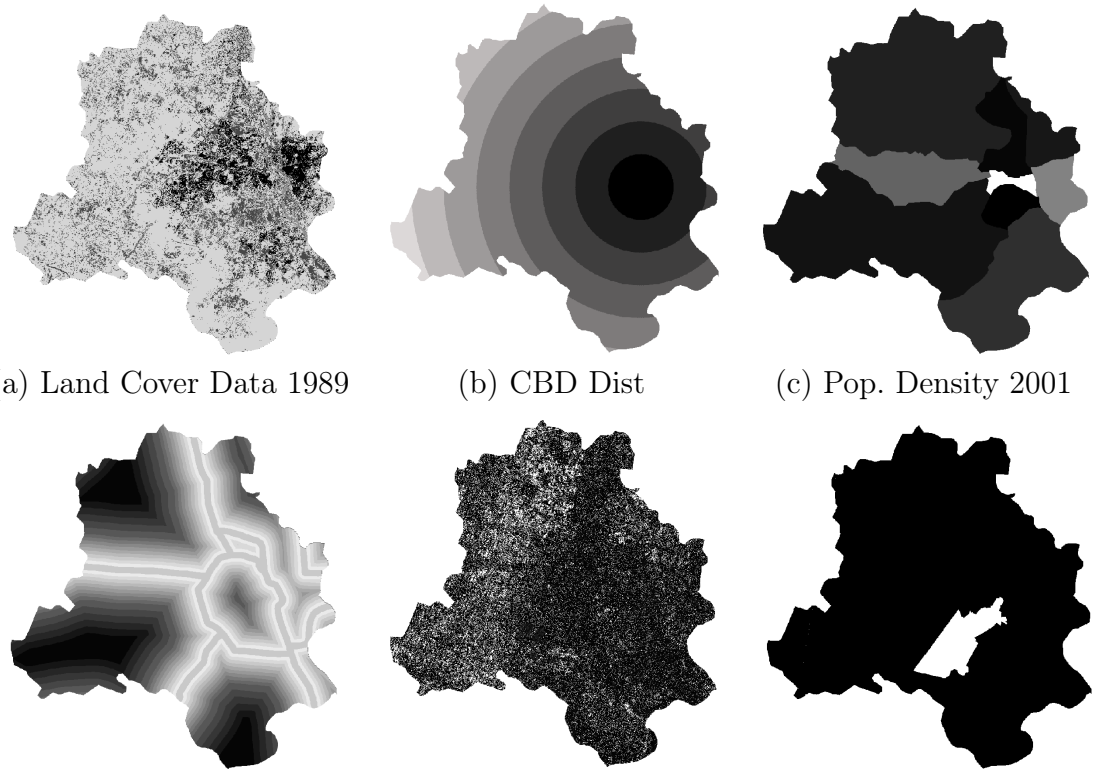
5.2 LUCC in Delhi, India CA-FTS

The complete processed data used in the study Tripathy and Kumar [2019] is available for download in Amit Kumar [2019]. As described in the previous chapter, the dataset consists of raster layers, shared as Geo-TIFF files, a public domain metadata embedded within an image format file mainly used for geo-referencing information. As detailed in chapter 3, in order to properly deal with these geography TIFF files, visualize the maps and artificially paint the layers, the QGIS platform was used. The raster layers collectively provide relevant information about land cover and land use changes over time in urban sprawl, which form the set of input variables for the CA-FTS proposed system, similar to the original work. Therefore, the variables used to build the model are:

1. Actual data from 1989, 1994, 1999, 2004, and 2014 based on imagery satellite, with four classes: Build-up, Water, Vegetation, and Others;
2. CBD distance map with seven levels of distance;
3. Population density maps for years 1991, 2001, 2011, and projections for 2019 and 2024 with nine levels of density;
4. Road Distance map with sixteen levels of distance;
5. Slope map with three degrees;
6. DDA Restricted Area map.

The input variables described are recalled in Figure 23. Those maps were already presented in Figure 14 and 15 in the previous chapter when explaining the original work

Figure 23 – Image-based input variables for the CA-FTS model development



(a) Land Cover Data 1989 (b) CBD Dist (c) Pop. Density 2001

(e) Road distance (f) Slope (g) Restricted area

approach. It is relevant to note that in Figure 23 (a) only the first year's land cover map is presented, although the other years mentioned were also used.

The pyFTS library offers a variety of data transformations used for data pre or post-processing. In this sense, the Differential transformation was used in the variables' data since they had different classes of representation. In order to illustrate it, in the population density map, the color symbology has a range that goes from 7.500 to 11.000, i. e., the colors in this map represent a value of thousands of units. On the other hand, for the slope map, the color symbology represents degrees 1, 2, and 3 units. Therefore, data pre-processing was necessary in order to equalize the data used for training and testing the model.

The main strategy for image-based input variables considers the grid of pixels of the image as the CA lattice. A single pixel represents a CA cell and its states are represented by the respective pixel's color - and as explained earlier, each color in the map characterizes a class of information. The images must have the same size in order to guarantee that the pixels in different maps and instants match at the exact location. Each of the maps, representing the variables shown in Figure 23, have 1386×820 pixels. Then, the proposed methodology is applied, similarly to the previous epidemiological model.

Building the FTS model starts by considering the training dataset, composed of the CA rules extracted from the images. The rules were formed based on Von Neumann's

neighborhood from the land cover (LC) map and the exogenous variables (EV). Let $F(t-1)$ be the fuzzified observation at time $(t-1)$ and $F_C(t)$ the center cell fuzzified observation at time (t), the Eq (3.1), for an order 1 model, can be rewritten as:

$$F_{LC}(t-1), F_{EV}(t-1) \rightarrow F_C(t) \tag{5.1}$$

where,

$$F_{LC}(t-1) = [F_N(t-1), F_S(t-1), F_W(t-1), F_E(t-1), F_C(t-1))] \tag{5.2}$$

$$F_{EV}(t-1) = [F_{CBD}(t-1), F_{PD}(t-1), F_{RD}(t-1), F_{SP}(t-1), F_{RA}(t-1)] \tag{5.3}$$

Considering North (N), South (S), West (W), East (E), Center (C), Central Business District (CBD), Population density (PD), Road distance (RD), Slope (SP) and Restricted area (RA). In other words, for every cell in the image, the rule is formed by extracting the neighborhood cell's state, the central cell state, and the additional exogenous variables for the central cell. Once the extracting process finishes for the training year image, the complete training data is ready for use by the pyFTS model.

Due to the irregular pixel distribution of the four interest features, the study case utilized the Multivariate FTS (MVFTS) rather than the Weighted version. Figure 24 presents the percentage of the features per year of the land cover maps. It can be noted that the River represents only around 0.50% of the pixels, thus, the weighted method would inappropriately act in favor of the other features.

Although using pyFTS library to build a model is not, in general, a high-cost processing task, a workaround was necessary to overcome memory limitations, running the algorithm in a simple personal computer machine (PC) - Intel Core i5 CPU 1.60GHz/2.11 GHz, RAM 8GB. It consists in dividing the image into four quadrants and building four MVFTS models for each. Since fewer data is processed, the processing cost is lower in comparison to creating a single model for the whole image and causing memory problems. The four models follow the same methodology to be constructed, the only change is the pixel information within each space. They together embody the model for LUCC CA-FTS forecasting simulation. Figure 25 exemplifies the described strategy.

As mentioned in the earlier section, the training approach for Delhi CA-FTS simulation considers only one land cover map from the first year available (1989), along with the exogenous variables, to extract the rules. Due to a large number of pixels, it was possible to compose a diverse set of rules using only one data map. The process of fitting

Figure 24 – Pixels distribution per feature for years 1999, 2004, 2009, 2014, 2019, and 2024.

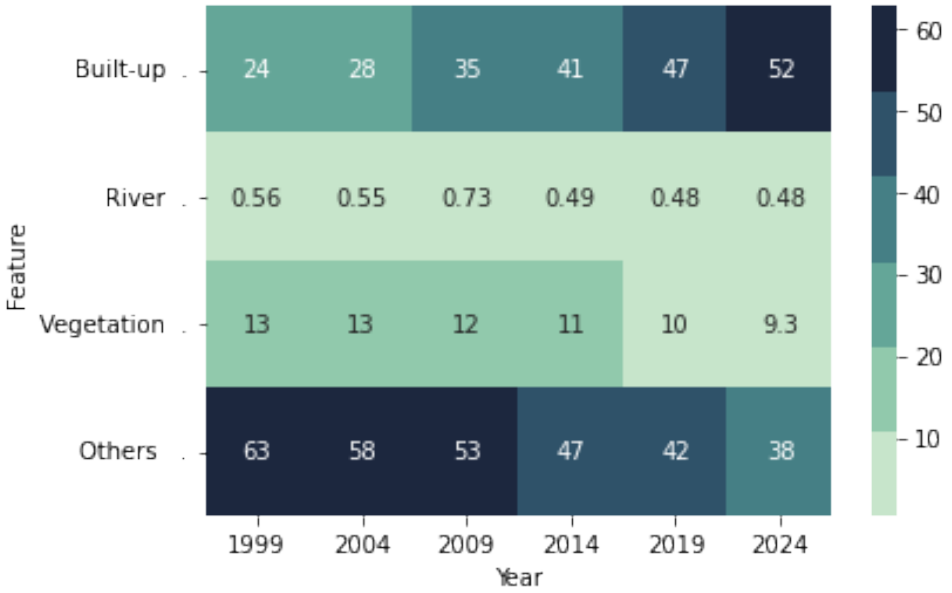
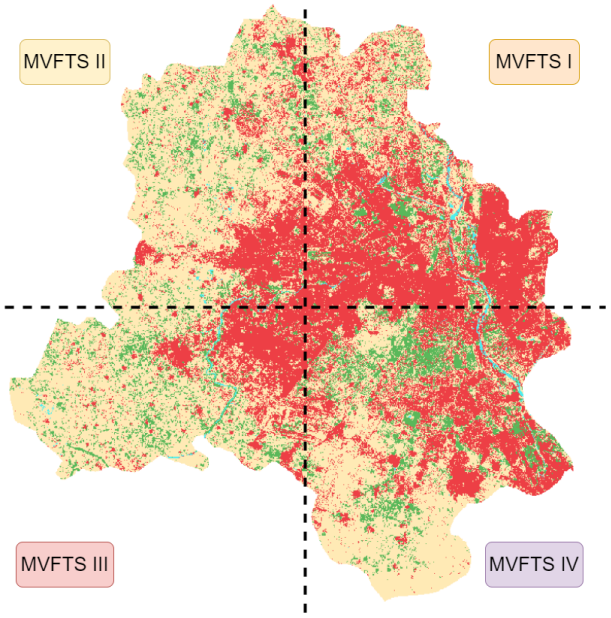


Figure 25 – Four quadrants division and its associated models



the model takes around two minutes and as a result, a completely trained forecast model is produced. At this point, the model is able to be applied in the CA simulation to predict future cell states. The year 1989 was used to train the model (extracting the pixel patterns along the image), then the following years composes the test dataset and used to validate it, providing an unbiased evaluation. Thus, firstly the model is executed for each pixel from the 1989 image map (t), forming the predicted 1999 map ($t + 1$). This procedure is continually made for the following years.

The model was analyzed following two configurations related to the data transfor-

mation applied in the pre-processed phase: the first one it was used BoxCox and the second the Differential, both available in the pyFTS library. In this way, the QGIS platform was also used to deal with the files and obtain the visualization and calculate the evaluation metrics. The system variables were analyzed considering 6 fuzzy sets.

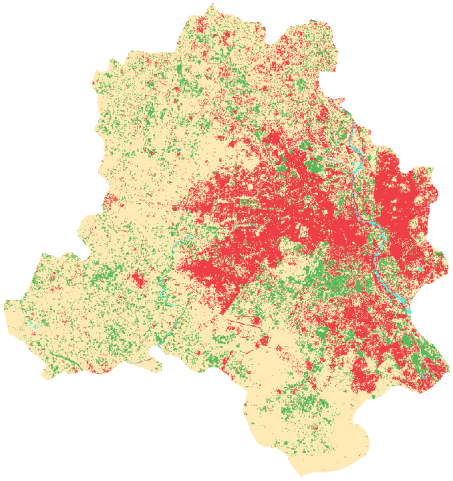
5.2.1 Results and CA-FTS Model Analysis

The first results considered BoxCox transformation in all the variables and it is presented in Figure 26 for the years 1999 and 2004, on the left is the Actual satellite-based map and on the right is the obtained CA-FTS forecasting output. Pos-processed and artificially colored in QGIS, the red color represents the Built-up feature, in green the Vegetation, in light blue the Water, and in light yellow the Others. A primarily visual analysis indicates the idea that the model could capture the principal characteristics of Delhi land in both years based on the previous year's information. The most notable error apparently was over the vegetation and built-up feature. Even the Water flag corresponding to $0,5\%$ of the pixels in the Actual image was preserved in the model output predicted map, located in East Delhi. In South West Delhi, which represents a region with a lower concentration of man-made features, both maps are very close, suggesting there is less error when dealing with places with this specificity. Figure 27 exhibits the actual and predicted results for the following years: 2009, 2014, 2019, and 2024.

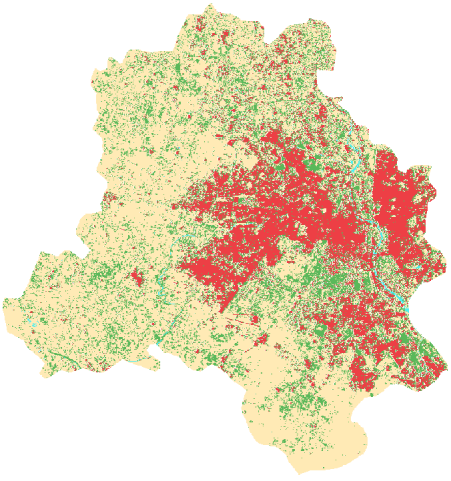
The evaluation metrics described in chapter 3 for map-based input were calculated in a way to quantify the differences between the expected image (actual data) and the image obtained by the CA-FTS model (predicted data). The Producer's Accuracy (PA) and User's Accuracy (UA) for each year and feature were represented using a heat map from the well-known Python library for data visualization Seaborn. Figure 28 displays both metrics results (a) for PA and (b) for UA. First, considering the producer's perspective, it is possible to observe that the Overall accuracy was an average of 80% for all the analyzed years, i. e., 80% of the pixels were correctly classified when comparing the expected results in the land cover map and the results obtained in the predicted one. The River feature, the one with very few pixels, had lower right classified rates, and Built-up, Vegetation, and Others had similar rate values. The User's perspective put the predicted map in front to compare with the land cover map backward, in order to check if the classified pixel matches the reference. Although there were missed pixels noted in PA's heat map, in River, all of the predicted pixels were indeed River in the reference. A point of attention was in 2014 when in User's perspective map had a lower accuracy when compared with others. Vegetation also represented a different behavior for years 2019 and 2024, most of them were right classified (UA) but the model missed around 40% of the expected in reference.

Figure 26 – Actual satellite-based data LUCC maps and its respectively CA-FTS forecasting result of Delhi for the years 1999 and 2004 with BoxCox Transformation

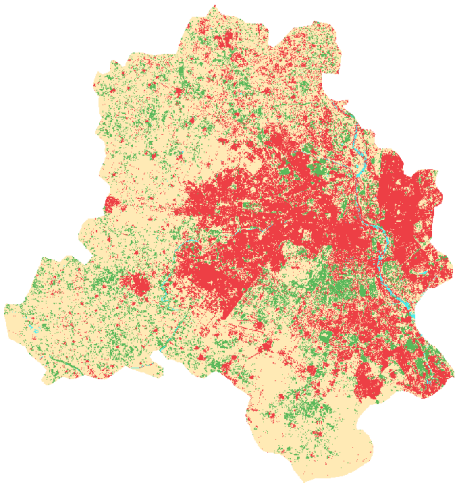
(a) Actual 1999



(b) Predicted 1999



(e) Actual 2004



(f) Predicted 2004

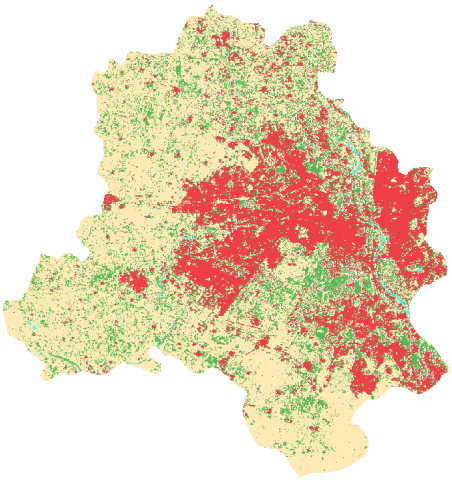
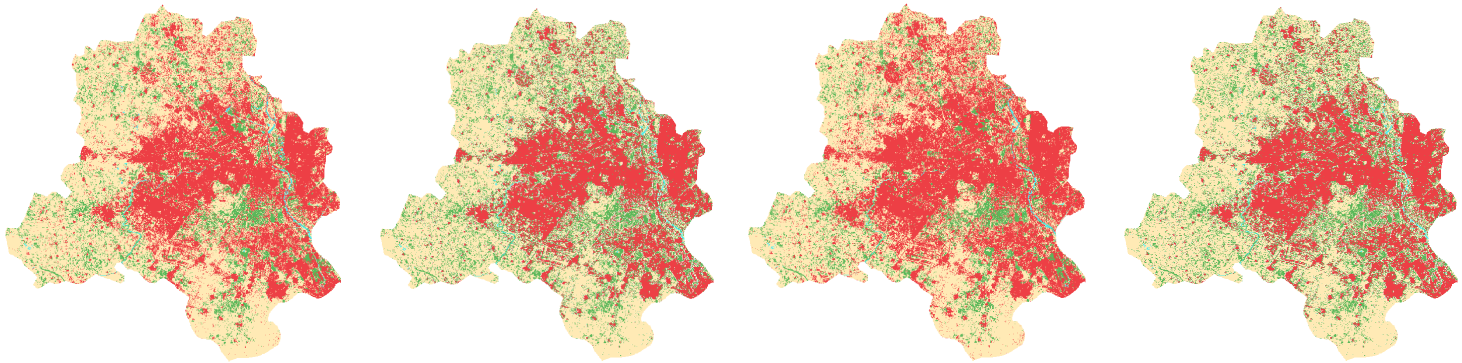


Figure 27 – Actual satellite-based data LUCC maps and its respectively CA-FTS forecasting result of Delhi for the years 2009, 2014, 2019 and 2024 with BoxCox Transformation

(a) Actual 2009 (b) Predicted 2009 (c) Actual 2014 (d) Predicted 2014



(e) Actual 2019 (f) Predicted 2019 (g) Actual 2024 (h) Predicted 2024

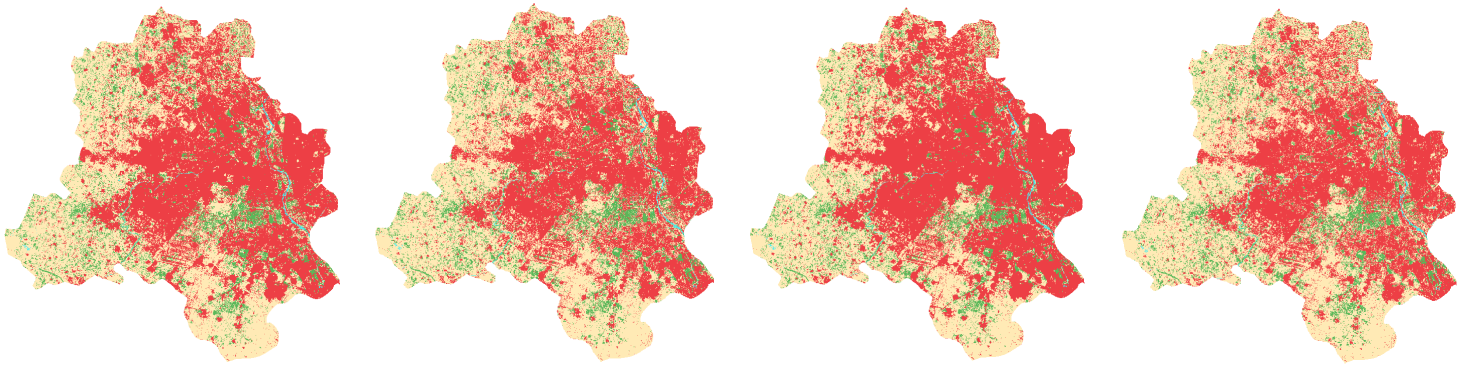
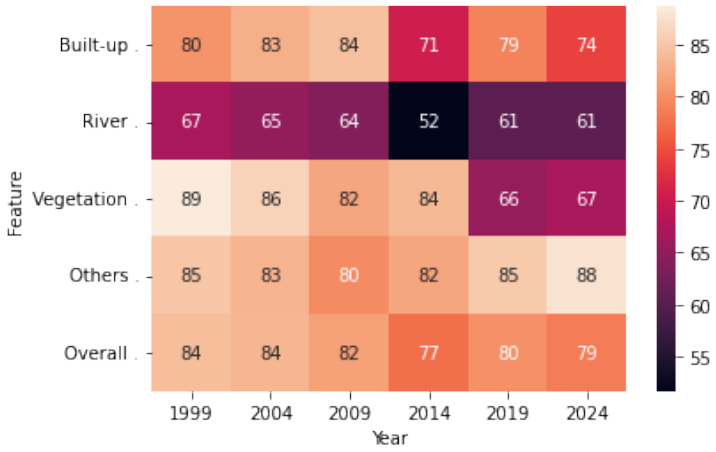
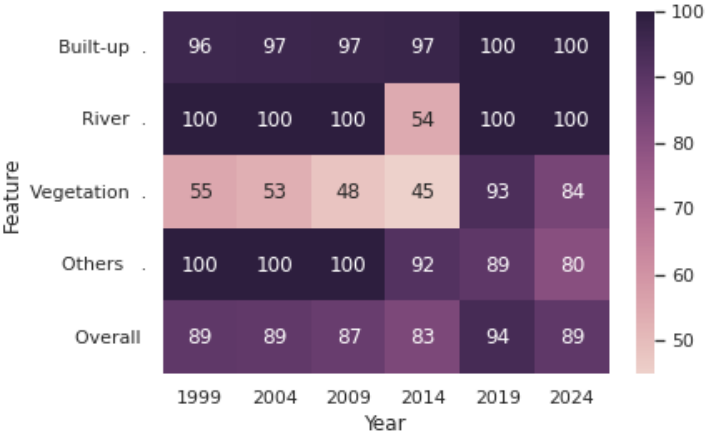


Figure 28 – Accuracy percentages with 6 fuzzy sets with BoxCox Transformation

(a) Producer's Accuracy



(b) User's Accuracy

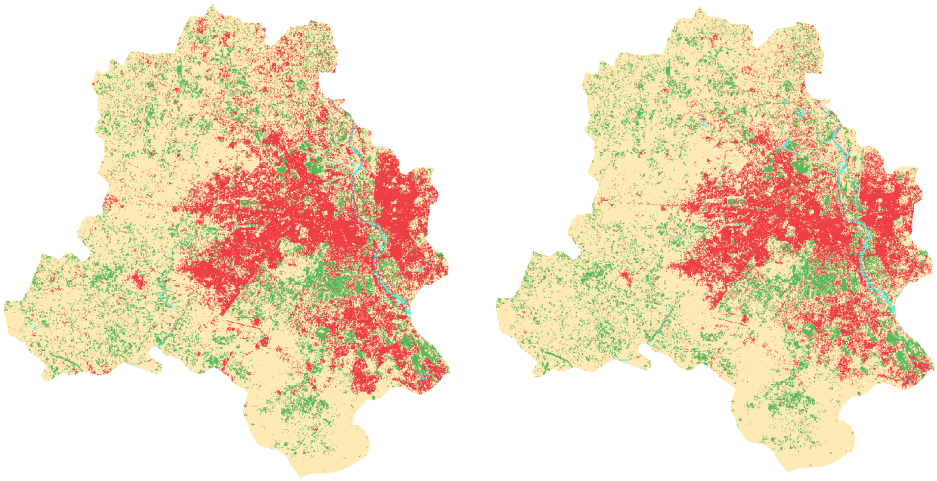


The next testing configuration analyzed considered was using Differential Transformation into the dataset as preprocessing. It is notable the similarity between the actual and the predicted map, as presented in Figure 29 for the years 1999 and 2004. In the South West and West Delhi the predicted map it is possible to see some errors around the Built-up class. The green class is hard to visualize the differences, being necessary a quantitative notion although suggesting a good fit. The Water, as already obtained in the last configuration, could be captured even having a low percentage of pixels classified in comparison with the three others. The following years - 2009, 2014, 2019, and 2024 - are presented lastly in Figure 30. Overall, the growth pattern behavior was reproduced based on the first-year configuration and additional information to predict a future behavior $(t+1)$ given a grid composition at time (t).

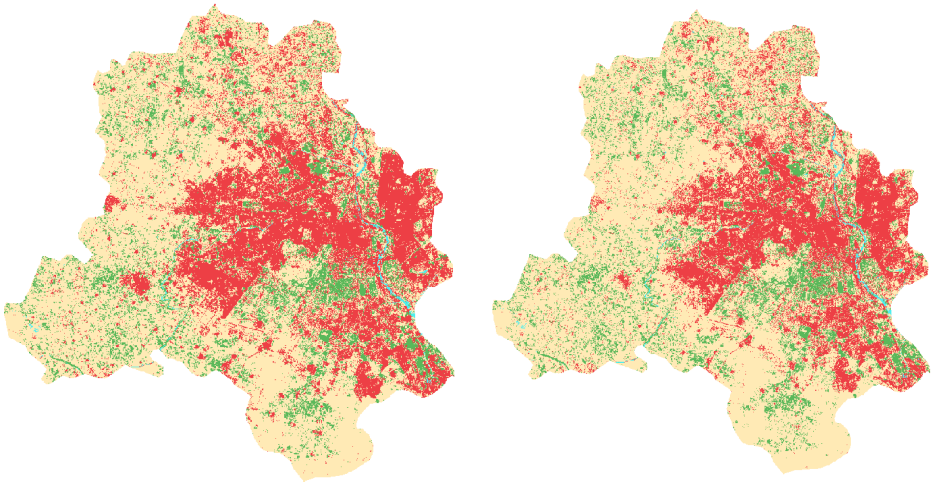
The same, previously used, evaluation metrics methodology was calculated for the new set-up configuration and exhibited in Figure 31. Considering the producer perspective, around 80% of the Built-up pixels (red) in the Reference map matches with the Predicted map. This means that, overall, 20% of the pixels were wrongly classified as a different class

Figure 29 – Actual satellite-based data LUCC maps and its respectively CA-FTS forecasting result of Delhi for the years 1999 and 2004 with Differential Transformation

(a) Actual 1999 (b) Predicted 1999



(e) Actual 2004 (f) Predicted 2004



than Built-up, the expected one. The River feature for years 2009 and 2014 had a lower accuracy percentage, indicating that something relevant may have happened in those years that wasn't taken into account in the system variables and the model fit. Following the same line of thought, all of the pixels (100%) for Vegetation and Others are paired when comparing the reference map for the analyzed years, i. e., whenever the reference map was expecting Vegetation and Others, the predicted map output the respective class. It demands to evaluate the other perspective's accuracy to validate the percentage of pixels classified as both classes in the Predicted map but wasn't expecting them in Reference one.

The UA heat map offers the option of obtaining the evaluation metrics from the angle of the Predicted Map. The first notable result is that the 100% of the Build-up classes in the Predicted map are indeed Build-up in the Reference map. For Vegetation and Others, an average of 90% was obtained. The 20% error in the Build-up feature in PA

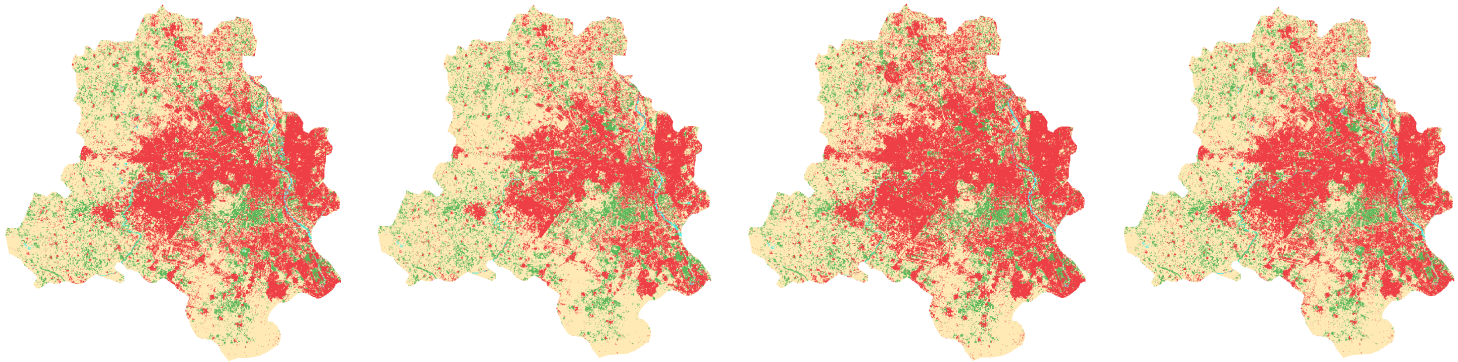
heat map was in. The lack in the years 2009 and 2014 was also identified in the UA map, reinforcing the thought of a missing relevant factor that wasn't considered by the model.

It is notable that the second CA-FTS model configuration had better metrics considering both approaches: qualitatively, through the map images, and quantitatively, with the PA and UA metrics in the heat map presented. It is relevant to mention that the model overfitting was avoided using only the first year as train data to build the MVFTS model and then applied it in the simulation for the following years.

In practice, the simulation execution indicated that the greater the number of cells in the CA lattice, the greater the time cost, once the pyFTS models need to go over each of the cells individually. In terms of image, for instance, the CA lattice can have thousands of cells (pixels). In terms of CAs simulations, this fact is expected and one of the biggest gains of using pyFTS methods is the lower time costing, taking from five to ten minutes to build the model with the earlier mentioned computer configuration.

Figure 30 – Actual satellite-based data LUCC maps and its respective CA-FTS forecasting result of Delhi for the years 2009, 2014, 2019 and 2024 with Differential Transformation

(a) Actual 2009 (b) Predicted 2009 (c) Actual 2014 (d) Predicted 2014



(e) Actual 2019 (f) Predicted 2019 (g) Actual 2024 (h) Predicted 2024

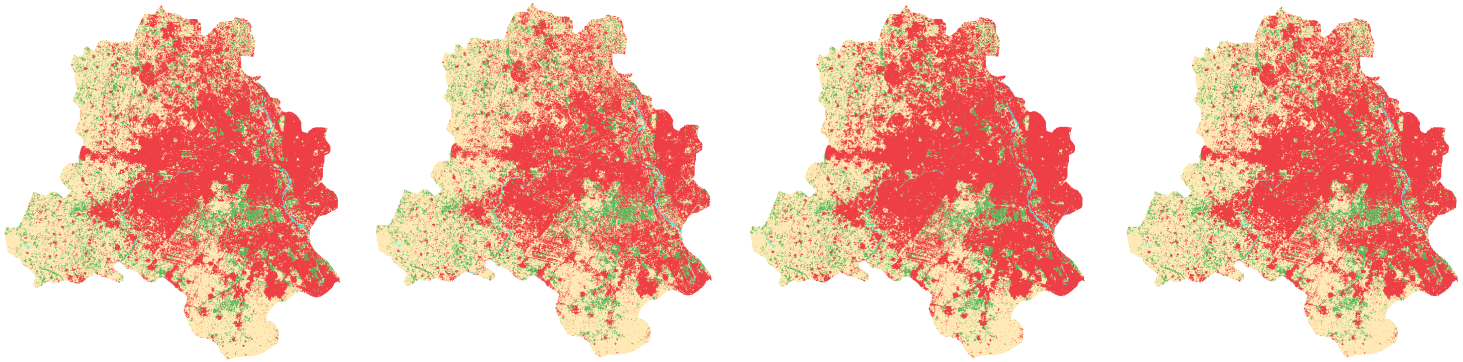
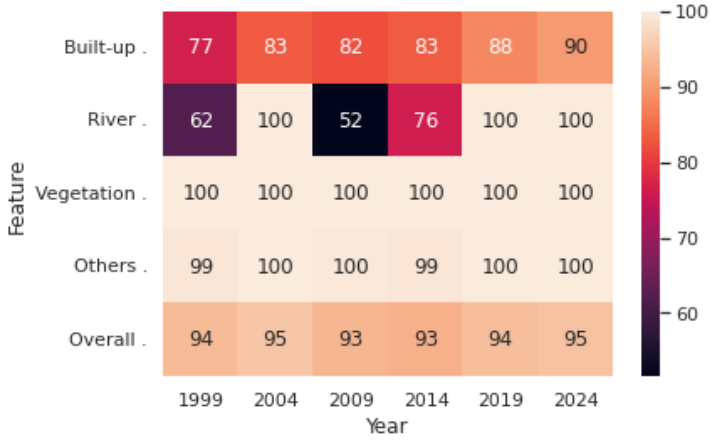
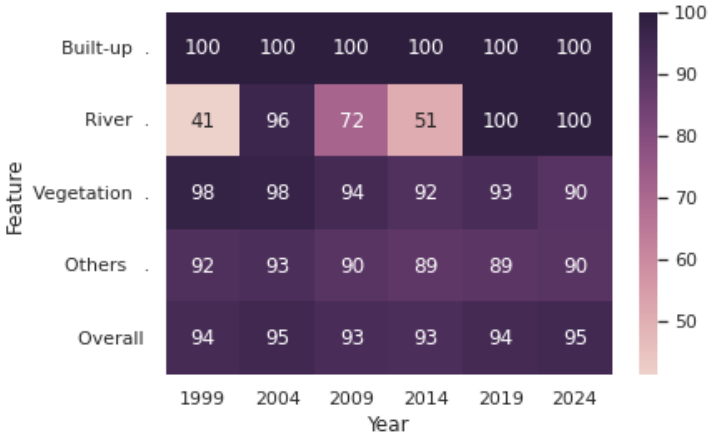


Figure 31 – Accuracy percentages with Differential Transformation

(a) Producer Accuracy



(b) User Accuracy



5.3 Discussion

The complexity in a multivariate model can be associated with the number of variables considered, once it increases the rule combinations Song [1999]. In this sense, it is notable that using pyFTS, using a basic personal computer configuration, represents a better option rather than using a metaheuristic search algorithm or genetic algorithm to calibrate a CA model. The proposed method, for both study cases, takes less than ten minutes to build the model, using the pyFTS library, and a fitness calculation of an optimization process can take hours, often used in a classic CA model for tuning the parameters transitional rules. Moreover, the proposed method can be used to model natural phenomena without the aid of advanced and complex studies, once the data-driven model is fully developed.

The strategy to overcome memory limitations, allocating four FTS models for each quadrant of an image, described in the Delhi case, demonstrated an interesting

aspect when using the pyFTS library. In others scenarios, one can choose different pyFTS methods for each region of an image, fitting a better model into the dataset peculiarities. The flexibility of the FTS framework is positive once it allows adaptation of the model parameters, meeting the problem specifications, in order to select good metrics results.

Although the CA-FTS model could extract information from historical images and build future scenarios, improvements need to be refined in order to consider different shapes on spatial distribution. In Delhi's case, using only one year as the training dataset, the model's rules contained the patterns of pixels for just a one time step. Improvements are necessary in order to consider other relevant information concerning climate change impacts, social economics dynamics, and others data to have a better spatio-temporal dynamic simulation. In both study cases, the obtained overall accuracy was greater than the Data-mining strategy, presented in Li and Yeh [2004], which had an average overall accuracy of 75%, in contrast to the around 95% using CA-FTS methodology.

The main advantage of CA-FTS methodology is the possibility of building the CA transitional rules without the need to infer and specify the rules for each study case, as traditionally CA simulations require. The rules are extracted from the data and variables that can be collected in several ways, as briefly discussed in Chapter 1. CA-FTS offers high accuracy and it can be used for spatio-temporal phenomena in which not much technical information is known, and once the data is available, the model can be constructed and the rules can be visualized and audited.

Chapter 6

Conclusions

Modeling real-world dynamical behavior is not a simple task and several (old and recent) strategies have been developed to simulate, classify and predict different types of scenarios. In the past, a set of differential equations were necessary to be calculated and defined in order to obtain a certain level of knowledge of a dynamical process. It is notable that the evolution of sciences and technologies opened new doors to overcome barriers and computers are important tools to support these types of simulations.

Cellular automata models are broadly used to simulate real-world dynamical systems. Its intrinsic discrete nature allows the reproduction of complex behaviors based on a set of simple rules, which can be complicated to be determined, requiring years of research, parameters calibration of optimization functions, or usage of Black-box AI algorithms. In this context, this work aimed to present a procedure to facilitate CA-based simulations, replacing the traditional set of transitional rules with a data-driven approach through fuzzy time series, named CA-FTS. FTS is a supervised algorithm forecasting framework and its flexibility is observed in a myriad of studies, as discussed in Chapter 2, and the integration with the CA perspective was conceptually proved in this work. In other words, CA-FTS can induce transition rules for a cellular automaton, reproducing the spatio-temporal dynamics extracted from the original dataset. Furthermore, the discovered rules are interpretable and can help understand the problem's underlying dynamics.

The detailed mathematical background that supports the CA-FTS method was presented and applied to two different study cases, an epidemiological model for Chagas disease spread and LUCC in Delhi, India. The model was constructed using the training dataset and validated with the test dataset. The obtained results indicated the potential of the CA-FTS modeling in the spatio-temporal dynamical simulation domain. Improvements to consider shapes, sizes, and rotation, for instance, are necessary to obtain a more accurate dynamic simulation, close to real-world scenarios. In terms of metrics, in the Chagas disease spread simulation, the normalized-RSME, and MAE had an average of 12% error. In LUCC Delhi studies, an overall accuracy of 90% was obtained, opposing the

75% in the Data-mining strategy proposed by Li and Yeh [2004], representing a significant improvement.

6.1 Future work

As a future project, probabilistic cellular automata can be developed in order to capture non-deterministic phenomena. Although it would include a parameter in the CA-FTS process, it can potentially improve certain types of real-world systems. The proposed methodology can also be evaluated in different types of datasets and processes. Since the current work is an initial work development, as a proof of concept, future work would involve the model comparison with traditional time series forecasting methods such as ARIMA, ARMA, SARIMA, and Holt-Winters, for instance.

In terms of application, there are numerous complex phenomena that can be applied and investigate CA-FTS abilities. Ultimately, the implementation of the CA-FTS approach in the pyFTS library will allow the method to be made accessible to the broader community.

6.2 Related Publications

In addition to the present work, a preliminary publication was generated focusing on the Chagas disease implementation:

ASTORE, L. M.; GUIMARÃES F. G. A.; JUNIOR, C. A. S. "Automatic Rule Generation for Cellular Automata using Fuzzy Times Series Methods" – 11th Brazilian Conference on Intelligent Systems (BRACIS) , 2022.

References

M. M. Aburas, Y. M. Ho, M. F. Ramli, and Z. H. Ash'aari. The simulation and prediction of spatio-temporal urban growth trends using cellular automata models: A review. *International Journal of Applied Earth Observation and Geoinformation*, 52:380–389, 2016. ISSN 1569-8432. doi: https://doi.org/10.1016/j.jag.2016.07.007. URL https://www.sciencedirect.com/science/article/pii/S0303243416301143.

P. T. Amit Kumar. Data for: Monitoring and modeling of spatio-temporal urban growth of delhi, india using cellular automata and geoinformatics. Mendeley Data, 2 2019. An optional note.

H. Balzter, P. W. Braun, and W. Köhler. Cellular automata models for vegetation dynamics. *Ecological Modelling*, 107(2):113–125, 1998. ISSN 0304-3800. doi: https://doi.org/10.1016/S0304-3800(97)00202-0. URL https://www.sciencedirect.com/science/article/pii/S0304380097002020.

S. M. Bartsch, J. K. Peterson, D. L. Hertenstein, L. Skrip, M. Ndeffo-Mbah, A. P. Galvani, A. P. Dobson, and B. Y. Lee. Comparison and validation of two computational models of chagas disease: A thirty year perspective from venezuela. *Epidemics*, 18:81–91, 2017. ISSN 1755-4365. doi: https://doi.org/10.1016/j.epidem.2017.02.004. URL https://www.sciencedirect.com/science/article/pii/S1755436516300743. Multi-model comparisons for neglected tropical diseases - validation and projection.

S. Basu and S. Basu. Different types of linear fuzzy cellular automata and their applications. *Fundam. Inf.*, 87(2):185–205, apr 2008. ISSN 0169-2968.

A. E. Bivona, A. S. Alberti, N. Cerny, S. N. Trinitario, and E. L. Malchiodi. Chagas disease vaccine design: the search for an efficient trypanosoma cruzi immune-mediated control. *Biochimica et Biophysica Acta (BBA) - Molecular Basis of Disease*, 1866(5): 165658, 2020. ISSN 0925-4439. doi: https://doi.org/10.1016/j.bbadis.2019.165658. URL https://www.sciencedirect.com/science/article/pii/S0925443919303898.

C. Bone, S. Dragicevic, and A. Roberts. A fuzzy-constrained cellular automata model of forest insect infestations. *Ecological Modelling*, 192(1):107–125, 2006. ISSN 0304-3800. doi:

https://doi.org/10.1016/j.ecolmodel.2005.09.013. URL https://www.sciencedirect.com/science/article/pii/S0304380005004771.

M. Bose and K. Mali. Designing fuzzy time series forecasting models: A survey. *International Journal of Approximate Reasoning*, 111:78–99, 2019. ISSN 0888-613X. doi: https://doi.org/10.1016/j.ijar.2019.05.002. URL https://www.sciencedirect.com/science/article/pii/S0888613X18306376.

L. Brugnano, F. Iavernaro, and P. Zanzottera. A multiregional extension of the SIR model, with application to the covid-19 spread in italy. 44(6):4414–4427, 2021. ISSN 0170-4214.

T. Carneiro. Nested-CA: A foundation for multiscale modeling of land use and land cover change. 01 2006.

G. Cattaneo, P. Flocchini, G. Mauri, C. Vogliotti, and N. Santoro. Cellular automata in fuzzy backgrounds. *Physica D: Nonlinear Phenomena*, 105(1):105–120, 1997. ISSN 0167-2789. doi: https://doi.org/10.1016/S0167-2789(96)00233-3. URL https://www.sciencedirect.com/science/article/pii/S0167278996002333.

Census. Primary census abstracts, registrar general of india, ministry of home affairs, government of india, 2011.

C. Chagas. Nova tripanozomiaze humana: estudos sobre a morfolojia e o ciclo evolutivo do schizotrypanum cruzi n. gen., n. sp., ajente etiolojico de nova entidade morbida do homem. *Memorias Do Instituto Oswaldo Cruz*, 1:159–218, 1909.

S. Chakraborty, H. Dadashpoor, J. Novotný, I. Maity, A. Follmann, P. P. Patel, U. Roy, and S. Pramanik. In pursuit of sustainability – spatio-temporal pathways of urban growth patterns in the world's largest megacities. *Cities*, 131:103919, 2022. ISSN 0264-2751. doi: https://doi.org/10.1016/j.cities.2022.103919. URL https://www.sciencedirect.com/science/article/pii/S0264275122003584.

C. Chatfield. *Time-Series Forecasting*. CRC Press, 2000. ISBN 9781420036206. URL https://books.google.com.br/books?id=PFHMBQAAQBAJ.

H. Che Ngoc, T. Vo-Van, Q.-C. Huynh-Le, H. Vu, T. Nguyen-Trang, and M.-T. Chu-Thi. *An Improved Fuzzy Time Series Forecasting Model*, pages 474–490. 01 2018. ISBN 978-3-319-73149-0. doi: 10.1007/978-3-319-73150-6_38.

C.-H. Cheng, T.-L. Chen, and C.-H. Chiang. Trend-weighted fuzzy time-series model for TAIEX forecasting. volume 4234, pages 469–477, 10 2006. ISBN 978-3-540-46484-6. doi: 10.1007/11893295_52.

B. Chopard and M. Droz. *Cellular Automata Modeling of Physical Systems*. 01 1998. doi: 10.1007/978-0-387-30440-3_57.

B. Chopard, A. Dupuis, A. MASSELOT, and P. LUTHI. Cellular automata and lattice boltzmann techniques: An approach to model and simulate complex systems. *Advances in Complex Systems (ACS)*, 05:103–246, 06 2002. doi: 10.1142/S0219525902000602.

M. K. Chowdary, J. Anitha, and D. J. Hemanth. Emotion recognition from EEG signals using recurrent neural networks. *Electronics*, 11(15), 2022. ISSN 2079-9292. URL https://www.mdpi.com/2079-9292/11/15/2387.

F. A. Chyon, M. N. H. Suman, M. R. I. Fahim, and M. S. Ahmmed. Time series analysis and predicting COVID-19 affected patients by ARIMA model using machine learning. *Journal of Virological Methods*, 301:114433, 2022. ISSN 0166-0934. doi: https://doi.org/10.1016/j.jviromet.2021.114433. URL https://www.sciencedirect.com/science/article/pii/S0166093421003724.

K. Clarke, S. Hoppen, and G. J. A self-modifying cellular automaton model of historical urbanization in the San Francisco bay area. *Environment and Planning B: Planning and Design*, 24:247–261, 02 1997. doi: 10.1068/b240247.

J. Conway. The game of life. *Scientific American*, 223(4):4, 1970. URL https://www.scopus.com/inward/record.uri?eid=2-s2.0-84924933935&partnerID=40&md5=b6f44f36c7fa870548eab80106590bf9. Cited by: 27.

W. Costa, L. Medeiros, and S. Sandri. A fuzzy cellular automata for SIR compartmental models. 11 2013. ISBN 978-3-319-03199-6. doi: 10.1007/978-3-319-03200-9_24.

J. Czerniak, H. Zarzycki, L. Apiecionek, W. Palczewski, and P. Kardasz. A cellular automata-based simulation tool for real fire accident prevention. *Mathematical Problems in Engineering*, 2018:1–12, 02 2018. doi: 10.1155/2018/3058241.

P. C. de Lima e Silva. pyfts: Fuzzy time series for python. 2018. doi: 10.5281/zenodo.597359.

P. C. de Lima e Silva. *Scalable Models for Probabilistic Forecasting with Fuzzy Time Series*. PhD thesis, 09 2019.

P. C. de Lima Silva, H. J. Sadaei, R. Ballini, and F. G. Guimarães. Probabilistic forecasting with fuzzy time series. *IEEE Transactions on Fuzzy Systems*, 28(8):1771–1784, 2020. doi: 10.1109/TFUZZ.2019.2922152.

S. Deep and A. Saklani. Urban sprawl modeling using cellular automata. *The Egyptian Journal of Remote Sensing and Space Science*, 17(2):179–187, 2014. ISSN 1110-9823. doi: https://doi.org/10.1016/j.ejrs.2014.07.001. URL https://www.sciencedirect.com/science/article/pii/S1110982314000180.

R. u. Din and E. A. Algehyne. Mathematical analysis of covid-19 by using SIR model with convex incidence rate. *Results in physics*, 23:103970–103970, 2021. ISSN 2211-3797.

C. Diniz. *Três Décadas de Mudanças na Planície Costeira Brasileira: O Status dos Manguezais, da Aquicultura e Salicultura a Partir de Séries Temporais Landsat e Técnicas de Aprendizado de Máquina*. PhD thesis, 2021.

D. Dominguez, L. d. J. del Villar, O. Pantoja, and M. González-Rodríguez. Forecasting amazon rain-forest deforestation using a hybrid machine learning model. *Sustainability*, 14(2), 2022. ISSN 2071-1050. URL https://www.mdpi.com/2071-1050/14/2/691.

R. Efendi, Z. Ismail, and M. M. Deris. A new linguistic out-sample approach of fuzzy time series for daily forecasting of malaysian electricity load demand. *Applied Soft Computing*, 28:422–430, 2015. ISSN 1568-4946. doi: https://doi.org/10.1016/j.asoc.2014.11.043. URL https://www.sciencedirect.com/science/article/pii/S1568494614006073.

Y. Feng, Y. Liu, and M. Batty. Modeling urban growth with GIS based cellular automata and least squares SVM rules: a case study in Qingpu–Songjiang area of Shanghai, China. *Stochastic environmental research and risk assessment*, 30(5):1387–1400, 2016. ISSN 1436-3240.

E. M. Forney. Electroencephalogram classification by forecasting with recurrent neural networks. 2011.

L. M. Fraga, G. M. B. de Oliveira, and L. G. A. Martins. Multistage evolutionary strategies for adjusting a cellular automata-based epidemiological model. In *2021 IEEE Congress on Evolutionary Computation (CEC)*, pages 466–473, 2021. doi: 10.1109/CEC45853.2021.9504738.

S. C. Fu and G. Milne. Epidemic modelling using cellular automata. *ACAL2003: The First Australian Conference on Artificial Life; Canberra, Australia*, 01 2003.

M. Gardner. Mathematical games. *Scientific American*, 223(4):120–123, 1970. ISSN 00368733, 19467087. URL http://www.jstor.org/stable/24927642.

A. Gharaibeh, A. Shaamala, R. Obeidat, and S. Al-Kofahi. Improving land-use change modeling by integrating ann with cellular automata-Markov chain model. *Heliyon*, 6: e05092, 09 2020. doi: 10.1016/j.heliyon.2020.e05092.

P. Ghosh, A. Mukhopadhyay, A. Chanda, P. Mondal, A. Akhand, S. Mukherjee, S. Nayak, S. Ghosh, D. Mitra, T. Ghosh, and S. Hazra. Application of cellular automata and Markov-chain model in geospatial environmental modeling - A review. *Remote Sensing Applications: Society and Environment*, 5:64–77, 01 2017. doi: 10.1016/j.rsase.2017.01.005.

M. Hassan and S. Elhassan. Modelling of urban growth and planning: A critical review. *Journal of Building Construction and Planning Research*, 08:245–262, 01 2020. doi: 10.4236/jbcpr.2020.84016.

H. Hethcote. The mathematics of infectious diseases. 42:599–653., 01 2000.

K. Huarng and H.-K. Yu. A type 2 fuzzy time series model for stock index forecasting. *Physica A: Statistical Mechanics and its Applications*, 353:445–462, 2005. ISSN 0378-4371. doi: https://doi.org/10.1016/j.physa.2004.11.070. URL https://www.sciencedirect.com/science/article/pii/S0378437105000725.

N. F. Ibrahim and X. Wang. Decoding the sentiment dynamics of online retailing customers: Time series analysis of social media. *Computers in Human Behavior*, 96:32–45, 2019. ISSN 0747-5632. doi: https://doi.org/10.1016/j.chb.2019.02.004. URL https://www.sciencedirect.com/science/article/pii/S0747563219300561.

H. Inaba and H. Sekine. A mathematical model for chagas disease with infection-age-dependent infectivity. *Mathematical Biosciences*, 190(1):39–69, 2004. ISSN 0025-5564. doi: https://doi.org/10.1016/j.mbs.2004.02.004. URL https://www.sciencedirect.com/science/article/pii/S0025556404000732.

Z. Ismail and R. Efendi. Enrollment forecasting based on modified weight fuzzy time series. *Journal of Artificial Intelligence*, 4, 01 2011. doi: 10.3923/jai.2011.110.118.

J. Kari. Theory of cellular automata: A survey. *Theoretical Computer Science*, 334 (1):3–33, 2005. ISSN 0304-3975. doi: https://doi.org/10.1016/j.tcs.2004.11.021. URL https://www.sciencedirect.com/science/article/pii/S030439750500054X.

X. Li and X. Liu. An extended cellular automaton using case-based reasoning for simulating urban development in a large complex region. *International Journal of Geographical Information Science*, 20:1109–1136, 11 2006. doi: 10.1080/13658810600816870.

X. Li and A. G.-O. Yeh. Data mining of cellular automata's transition rules. *International Journal of Geographical Information Science*, 18(8):723–744, 2004. doi: 10.1080/13658810410001705325.

Y. Li, M. Chen, Z. Dou, X. Zheng, Y. Cheng, and A. Mebarki. A review of cellular automata models for crowd evacuation. *Physica A: Statistical Mechanics and its Applications*, 526: 120752, 2019. ISSN 0378-4371. doi: https://doi.org/10.1016/j.physa.2019.03.117. URL https://www.sciencedirect.com/science/article/pii/S0378437119303528.

I. Lima and P. P. Balbi. Estimates of the collective immunity to covid-19 derived from a stochastic cellular automaton based framework. *Natural Computing*, 21:449–461, 2022. ISSN 1572-9796. doi: https://doi.org/10.1007/s11047-022-09893-3.

Y. Liu and S. Phinn. Modeling urban development with cellular automata incorporating fuzzy-set approaches. *Computers, Environment and Urban Systems*, 27:637–658, 11 2003a. doi: 10.1016/S0198-9715(02)00069-8.

Y. Liu and S. Phinn. Modeling urban development with cellular automata incorporating fuzzy-set approaches. *Computers, Environment and Urban Systems*, 27:637–658, 11 2003b. doi: 10.1016/S0198-9715(02)00069-8.

L. Mantelas, P. Prastacos, T. Hatzichristos, and K. Koutsopoulos. Using fuzzy cellular automata to access and simulate urban growth. *GeoJournal*, 77:13–28, 02 2012. doi: 10.1007/s10708-010-9372-8.

K. Małecki. Graph cellular automata with relation-based neighbourhoods of cells for complex systems modelling: A case of traffic simulation. *Symmetry*, 9:322, 12 2017. doi: 10.3390/sym9120322.

G. Melotti. Application of cellular automata in complex systems: A study of case in spreading epidemics. Master's thesis, Federal University of Minas Gerais, Minas Gerais, Brazil, 2 2009.

I. A. Moosa. *Univariate Time Series Techniques*, pages 62–97. Palgrave Macmillan UK, London, 2000. ISBN 978-0-230-37900-8. doi: 10.1057/9780230379008_3. URL https://doi.org/10.1057/9780230379008_3.

K. Nagel and M. Schreckenberg. A cellular automaton model for freeway traffic. *Journal de Physique I*, 2:2221, 12 1992. doi: 10.1051/jp1:1992277.

U. Nations. Population 2030 demographic challenges and opportunities for sustainable development planning (st/esa/ser.a/389), 2015.

J. Neumann and A. Burks. Theory of self-reproducing automata. *Champaign, USA: University of Illinois Press*, 1966.

M. O'Connor, W. Remus, and K. Griggs. Judgemental forecasting in times of change. *International Journal of Forecasting*, 9(2):163–172, 1993. ISSN 0169-2070. doi: https://doi.org/10.1016/0169-2070(93)90002-5. URL https://www.sciencedirect.com/science/article/pii/0169207093900025.

G. Oliveira, P. P. B. de Oliveira, and N. Omar. Definition and application of a five-parameter characterization of one-dimensional cellular automata rule space. *Artificial Life*, 7:277–301, 07 2001. doi: 10.1162/106454601753238645.

D. Parker, S. Manson, M. Janssen, M. Hoffmann, and P. Deadman. Multi-agent systems for the simulation of land-use and land-cover change: A review. *Annals of the Association of American Geographers*, 93:314 – 337, 06 2003. doi: 10.1111/1467-8306.9302004.

F. Pereira and P. Schimit. Dengue fever spreading based on probabilistic cellular automata with two lattices. *Physica A: Statistical Mechanics and its Applications*, 499:75–87,

2018. ISSN 0378-4371. doi: https://doi.org/10.1016/j.physa.2018.01.029. URL https://www.sciencedirect.com/science/article/pii/S0378437118300499.

B. Pijanowski, D. Brown, B. Shellito, and G. Manik. Using neural networks and gis to forecast land use changes: A land transformation model. *Computers, Environment and Urban Systems*, 26:553–575, 11 2002. doi: 10.1016/S0198-9715(01)00015-1.

J. A. Pérez-Molina and I. Molina. Chagas disease. *The Lancet*, 391(10115):82–94, 2018. ISSN 0140-6736. doi: https://doi.org/10.1016/S0140-6736(17)31612-4. URL https://www.sciencedirect.com/science/article/pii/S0140673617316124.

Y. Qiang and N. S. N. Lam. Modeling land use and land cover changes in a vulnerable coastal region using artificial neural networks and cellular automata. *Environmental Monitoring and Assessment*, 187(3), 2015. doi: 10.1007/s10661-015-4298-8. Cited by: 79.

S. Raschka. *Python Machine Learning*. Packt Publishing - ebooks Account, 2015. ISBN 1783555130.

L. Sang, C. Zhang, J. Yang, D. Zhu, and W. Yun. Simulation of land use spatial pattern of towns and villages based on CA–Markov model. *Mathematical and Computer Modelling*, 54(3):938–943, 2011. ISSN 0895-7177. doi: https://doi.org/10.1016/j.mcm.2010.11.019. URL https://www.sciencedirect.com/science/article/pii/S0895717710005108. Mathematical and Computer Modeling in agriculture (CCTA 2010).

I. Santé, A. M. García, D. Miranda, and R. Crecente. Cellular automata models for the simulation of real-world urban processes: A review and analysis. *Landscape and Urban Planning*, 96(2):108–122, 2010. ISSN 0169-2046. doi: https://doi.org/10.1016/j.landurbplan.2010.03.001. URL https://www.sciencedirect.com/science/article/pii/S0169204610000472.

P. Schimit. A model based on cellular automata to estimate the social isolation impact on covid-19 spreading in brazil. *Computer Methods and Programs in Biomedicine*, 200, 2021. ISSN 0169-2607. doi: https://doi.org/10.1016/j.cmpb.2020.105832.

C. A. Severiano, P. C. de Lima e Silva, M. Weiss Cohen, and F. G. Guimarães. Evolving fuzzy time series for spatio-temporal forecasting in renewable energy systems. *Renewable Energy*, 171:764–783, 2021. ISSN 0960-1481. doi: https://doi.org/10.1016/j.renene.2021.02.117. URL https://www.sciencedirect.com/science/article/pii/S0960148121002962.

P. C. L. Silva, P. de Oliveira e Lucas, H. J. Sadaei, and F. G. Guimarães. Distributed evolutionary hyperparameter optimization for fuzzy time series. *IEEE Transactions on*

Network and Service Management, 17(3):1309–1321, 2020. doi: 10.1109/TNSM.2020.2980289.

R. Slimi, S. El Yacoubi, E. Dumonteil, and S. Gourbière. A cellular automata model for chagas disease. *Applied Mathematical Modelling*, 33(2):1072–1085, 2009. ISSN 0307-904X. doi: https://doi.org/10.1016/j.apm.2007.12.028. URL https://www.sciencedirect.com/science/article/pii/S0307904X07003563.

Q. Song. Seasonal forecasting in fuzzy time series. *Fuzzy Sets and Systems*, 107(2):235–236, 1999. ISSN 0165-0114. doi: https://doi.org/10.1016/S0165-0114(98)00266-8. URL https://www.sciencedirect.com/science/article/pii/S0165011498002668.

Q. Song and B. S. Chissom. Fuzzy time series and its models. *Fuzzy Sets and Systems*, 54 (3):269–277, 1993. ISSN 0165-0114. doi: https://doi.org/10.1016/0165-0114(93)90372-O. URL https://www.sciencedirect.com/science/article/pii/016501149390372O.

S. Suhartono, M. H. Lee, and H. SADAEI. A weighted fuzzy integrated time series for forecasting tourist arrivals. volume 252, pages 206–217, 11 2011. ISBN 978-3-642-25452-9. doi: 10.1007/978-3-642-25453-6_19.

C. Suárez, D. Nolder, A. García-Mingo, D. A. J. Moore, and P. L. Chiodini. Diagnosis and clinical management of chagas disease: An increasing challenge in non-endemic areas. *Research and Reports in Tropical Medicine*, 13:25–49, 2022. doi: 10.2147/RRTM.S278135.

A. Talukder. Susceptible-infectious-recovered (SIR) model-based forecasting of covid-19 outbreak in bangladesh. *International journal of clinical practice (Esher)*, 74(11): e13648–n/a, 2020. ISSN 1368-5031.

L. D. Tavares. An urban traffic simulator based on cellular automata. Master's thesis, Federal University of Minas Gerais, Minas Gerais, Brazil, 3 2010.

T. Toffoli and N. Margolus. *The Margolus neighborhood*, pages 119–138. 1987.

P. Tripathy and A. Kumar. Monitoring and modelling spatio-temporal urban growth of delhi using cellular automata and geoinformatics. *Cities*, 90:52–63, 2019. ISSN 0264-2751. doi: https://doi.org/10.1016/j.cities.2019.01.021. URL https://www.sciencedirect.com/science/article/pii/S0264275118303871.

B. Turner Ii, D. Skole, S. Sanderson, G. Fischer, L. Fresco, and R. Leemans. *Land-use and land-cover change. Science/Research plan.* Number 35 in IGBP Report. IGBP, 1995. HDP Report No. 7.

P. Verburg, K. Kok, R. Pontius, and A. Veldkamp. *Modeling Land-Use and Land-Cover Change*, pages 117–135. 08 2008. ISBN 978-3-540-32201-6. doi: 10.1007/3-540-32202-7_5.

M. Wabomba. Modeling and forecasting kenyan gdp using autoregressive integrated moving average (arima) models. *Science Journal of Applied Mathematics and Statistics*, 4:64, 01 2016. doi: 10.11648/j.sjams.20160402.18.

J. Wang, M. Bretz, M. A. A. Dewan, and M. A. Delavar. Machine learning in modelling land-use and land cover-change (lulcc): Current status, challenges and prospects. *Science of The Total Environment*, 822:153559, 2022. ISSN 0048-9697. doi: https://doi.org/10.1016/j.scitotenv.2022.153559.

S. R. Wayne A. Woodward, Bivin Philip Sadler. *Time Series for Data Science*. Chapman and Hall/CRC, New York, 1 edition, 2022. ISBN 9781003089070. doi: https://doi.org/10.1201/9781003089070.

S. Wolfram. Universality and complexity in cellular automata. *Physica D: Nonlinear Phenomena*, 10(1):1–35, 1984. ISSN 0167-2789. doi: https://doi.org/10.1016/0167-2789(84)90245-8. URL https://www.sciencedirect.com/science/article/pii/0167278984902458.

X. Xiao, S.-H. Shao, and K.-C. Chou. A probability cellular automaton model for hepatitis b viral infections. *Biochemical and Biophysical Research Communications*, 342(2):605–610, 2006. ISSN 0006-291X. doi: https://doi.org/10.1016/j.bbrc.2006.01.166. URL https://www.sciencedirect.com/science/article/pii/S0006291X0600249X.

Q. Yang, X. Li, and X. Shi. Cellular automata for simulating land use changes based on support vector machines. *Computers Geosciences*, 34(6):592–602, 2008. ISSN 0098-3004. doi: https://doi.org/10.1016/j.cageo.2007.08.003. URL https://www.sciencedirect.com/science/article/pii/S0098300407002087.

A. Yeh and X. Li. Simulation of development alternatives using neural networks, cellular automata, and gis for urban planning. *Photogrammetric Engineering Remote Sensing*, 69:1043–1052, 09 2003. doi: 10.14358/PERS.69.9.1043.

H.-K. Yu. Weighted fuzzy time series models for TAIEX forecasting. *Physica A: Statistical Mechanics and its Applications*, 349(3):609–624, 2005. ISSN 0378-4371. doi: https://doi.org/10.1016/j.physa.2004.11.006. URL https://www.sciencedirect.com/science/article/pii/S0378437104014128.

L. Zadeh. Fuzzy sets. *Information and Control*, 8(3):338–353, 1965. ISSN 0019-9958. doi: https://doi.org/10.1016/S0019-9958(65)90241-X. URL https://www.sciencedirect.com/science/article/pii/S001999586590241X.

Z. Zheng, W. Huang, S. Li, and Y. Zeng. Forest fire spread simulating model using cellular automaton with extreme learning machine. *Ecological Modelling*, 348:33–43, 01 2017. doi: 10.1016/j.ecolmodel.2016.12.022.

R. Áurea Tonetti Massahud. Dengue propagation model using cellular automata. Master's thesis, Federal University of Lavras, Lavras, Brazil, 2 2011.