People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research
University of 8 May 1945-Guelma
Faculty of Mathematics, Computer Science and Science of Matter
Department of Computer Science

Master Thesis

Specialty: Computer Science

Option: Information Systems

Theme :

Intelligent Models for Dynamic Systems Monitoring: A Case Study on Forest Fires

---

Presented by : *SEDDIKI Loubna*

Jury Members:

- **Chairman** : Dr ZEDADRA Amina
- **Supervisor** : Dr BENHAMZA Karima
- **Examiner** : Dr DERDAR Salah
- **Representative POLE PRO** : Dr BERKAT Ghania

Juin 2024

Acknowledgements

*First and foremost, I express my sincere gratitude to the Almighty **Allah** for bestowing upon me the strength, energy, perseverance, and resilience to successfully complete this humble work. Alhamdulillah, I have finally reached this point and achieved one of my life's goals.*

*I extend my heartfelt thanks to Dr. **BENHAMZA Karima**, my supervisor, for her invaluable guidance and knowledge throughout this project. Her critical and innovative thinking has greatly influenced my approach to research, inspiring me to explore new ideas. I apologize for any inconvenience I may have caused and appreciate her patience and understanding,you are a remarkable role model, and I aspire to follow your dedication and passion in my future endeavors. I am truly grateful..*

*I express my heartfelt gratitude to the esteemed **members of the jury** for their invaluable presence and thorough examination of my thesis. I sincerely appreciate the time and effort they dedicated to carefully reviewing my work.*

*I would like to thank **Conservation Forestry Service of Guelma** for giving me the opportunity to conduct training with them. I am deeply grateful to Mr. **Chetaibi Mourad** Head of the Forest Resources Protection Department for providing valuable information that was essential in completing my work.*

*I would like to express my sincere gratitude to Pr. **BENKIRAT Abdelaziz** for his invaluable advice and assistance in this project startup. His unwavering support and insightful guidance have been instrumental in development this project .*

*I express my deep gratitude to Dr. **KEFALI Abderrahmane**, who was more than just a teacher . he was my mentor and support throughout my university years, and inspired my love for my field. His influence is the reason for my excellence in my studies. I'm really grateful.*

*I would also like to extend my heartfelt thanks to Dr. **Benrazek Ala Eddine** and **Lina** for their invaluable assistance in completing this work. Thank you for your support and for generously giving your valuable time. I am deeply grateful .*

Dedications

All praise is due to Allah, at the beginning and at the end. No exertion comes to completion, nor does any endeavor succeed, but by His elegance. With heartfelt love, I dedicate my master's thesis:

*To the one whose prayers were the secret of my success, to the one who was my light in my darkness, **Mom**, I dedicate this achievement to you. which would not have been possible without your sacrifices.*

*To my beloved **Dad** , whose unwavering support and boundless generosity have shaped the person I am today. You upheld me unequivocally and gave selflessly, never expecting anything in return. Your love and guidance have been my foundation and inspiration. Thank you.*

*To my idol and source of strength, whose spirit I miss dearly, my beloved **grandfather**, whom I always wished to have by my side. On this special day, I hope you are proud of me.*

*To my dear **sister** and **brother**, my immovable pillars of strength and quality. Your constant support and encouragement have been invaluable.*

*To my beloved friends **Islam** and **Rym**, your friendship has made this journey truly enjoyable and unforgettable. Your unwavering support and boundless joy have been invaluable throughout my master's studies. I am deeply grateful for your presence and everything you've done for me.*

To my family, for ceaselessly giving me with support and back.

Loubna

Abstract

Dynamic systems play a crucial role in various fields, including meteorology, finance, and technology, and are characterized by their complexity and interdependencies. Traditional modeling and prediction methods often struggle to capture the intricate behaviors and evolving patterns of these systems, leading to suboptimal control and prediction outcomes. Forest fires are a prime example of dynamic systems, where interactions among meteorological conditions, fuel types, and topography result in unpredictable and nonlinear fire spread patterns. This issue is particularly critical in regions like Algeria, where recent forest fires have caused significant damage, underscoring the need for advanced predictive and management tools.
This work aims to study dynamic systems and propose an intelligent and adaptive model for dynamic forest fire prediction using Deep Neural Networks (DNN) and Cellular Automata (CA). The primary advantage of this system lies in its ability to accurately predict fire ignition points based on meteorological and environmental data and to simulate fire spread across various landscapes with greater precision. This dual-method approach enhances detection and simulation accuracy, reduces response times for authorities, and improves wildfire containment and mitigation efforts.

Keywords: Complex systems; Forest Fire Management; Deep Neural Networks; Cellular Automata; Predictive Analytics, Fire Spread Simulation.

Résumé

Les systèmes dynamiques sont un aspect fondamental de divers domaines, influençant des secteurs tels que la météorologie, la finance et la technologie. Ces systèmes se caractérisent par leur complexité et leurs interdépendances, qui posent des défis importants aux méthodes traditionnelles de modélisation et de prévision. Les approches conventionnelles ne parviennent souvent pas à saisir les comportements complexes et les schémas évolutifs des systèmes dynamiques, ce qui se traduit par un contrôle et une prévision sous-optimaux. Les incendies de forêt sont des exemples de systèmes dynamiques, où les interactions entre les conditions météorologiques, les types de combustibles et la topographie entraînent des schémas de propagation des incendies imprévisibles et non linéaires. Il s'agit d'un problème critique, en particulier dans des régions comme l'Algérie, où les récents incendies de forêt ont causé des dégâts considérables, soulignant la nécessité de disposer d'outils de prévision et de gestion avancés. Notre objectif dans ce travail est de proposer un système intelligent et adaptatif de gestion des incendies de forêt en utilisant des réseaux neuronaux profonds et des automates cellulaires. Le principal avantage de notre système est sa capacité à prédire avec précision les points d'allumage des incendies sur la base de données météorologiques et environnementales et à simuler plus précisément la propagation des incendies dans différents paysages. Cette approche à double méthode améliore la précision de la détection et de la simulation, ce qui réduit les temps de réponse des autorités et améliore les efforts de confinement et d'atténuation des incendies de forêt.

Mots clés: Systèmes complexes; Gestion des incendies de forêt; Réseaux de neurones profonds; Automates cellulaires; Analyses prédictives; Simulation de propagation des incendies.

ملخص

الأنظمة الديناميكية هي جانب أساسي في مجالات متنوعة، تؤثر على قطاعات مثل الأرصاد الجوية، والمالية، والتكنولوجيا. تتميز هذه الأنظمة بتعقيدها وترابطها، مما يشكل تحديات كبيرة أمام أساليب النمذجة والتنبؤ التقليدية. غالبا ما تفشل الأساليب التقليدية في استيعاب السلوكيات المعقدة والأنماط التطورية للأنظمة الديناميكية، مما يؤدي إلى تحكم وتنبؤ غير مثاليين.
تعد حرائق الغابات مثال على الانظمة الديناميكية ، حيث تؤدي التفاعلات بين الظروف الجوية و التضاريس الى انشار حرائق غير متوقعة ، يشكل هذا مشكلة حاسمة في كثير من المناطق منها الجزائر، حيث تسببت حرائق الغابات الأخيرة في أضرار جسيمة، مما يبرز الحاجة إلى وجود أدوات متقدمة للتنبؤ والإدارة.
هدفنا في هذا العمل هو اقتراح نظام ذكي وقابل للتكيف لإدارة حرائق الغابات باستخدام الشبكات العصبية العميقة والخلايا التلقائية. يتمثل الفائدة الرئيسية لنظامنا في قدرته على التنبؤ بدقة بنقاط اشتعال الحرائق استنادا إلى البيانات الجوية والبيئية، ومحاكاة انتشار الحرائق بدقة أكبر في مختلف المناطق .
يعزز هذا النهج المزدوج من دقة الكشف والمحاكاة، مما يقلل من أوقات استجابة السلطات ويحسن جهود احتواء وتخفيف حرائق الغابات.

الكلمات المفتاحية الأنظمة المعقدة، إدارة حرائق الغابات، الشبكات العصبية العميقة، الخلايا التلقائية، التحليلات التنبؤية، محاكاة انتشار الحرائق

Contents

List of Tables

List of Figures

List of Abbreviations

DS Dynamic system

IA Artificial intelligence

ML Machine Learning

Dl Deep learning

RL Reinforcement Learning

DNN Deep Neural Network

DNNs Deep Neural Networks

DSP Dynamic Systems Predictions

ABMS Agent-based modeling and simulations

DT Decision Tree

RF Random Forest

SVM Support vector MAchine

LSTM long Short-Term Memory

FWI Fire Weather Index

FFMC Fine Fuel Moisture Code

DMC Duff Moisture Code

DC Drought Code

ISI Initial Spread Index

BUI Buildup Index

temp temperature

RH Relative humidity

Ws wind speed

General Introduction

Dynamic systems are integral to numerous aspects of modern life, influencing critical areas such as weather forecasting, financial market stability, and the functionality of technological systems. However, the inherent complexity and interdependencies within these systems pose significant challenges to traditional modeling and prediction methods. Conventional approaches often fail to capture the nuanced behaviors and evolving patterns of dynamic systems, leading to inaccuracies in forecasting future states and detecting anomalies in real-time. Addressing these challenges requires innovative methodologies that leverage cutting-edge technologies to enhance predictive capabilities and monitoring efficiency.

In recent years, forest fires have emerged as a pressing global concern, posing significant threats to ecosystems, wildlife, human safety, and infrastructure. Forest fires are dynamic systems due to their complex, evolving interactions with meteorological, fuel, and topographical factors. These interdependencies result in unpredictable and non-linear fire spread patterns. This escalating threat has become particularly pronounced in Algeria, a country with vast and diverse forested regions. Incidents like the devastating forest fire in Tizi Ouzou (2021), have left an indelible mark on both the landscape and the collective consciousness. Indeed, the Tizi Ouzou forest fire serves as a stark testament to the urgent need for advanced systems dedicated to monitoring and managing these dynamic environments, particularly in the context of prediction and management.

To overcome these limitations, this work proposes an intelligent system called FireTrack, which combines a Deep Neural Network (DNN) and Cellular Automata (CA) models to create an advanced approach for forest fire management. Proposed DNN model is employed to accurately predict ignition points using meteorological and environmental data, while CA model simulates the spread of fire across the landscape. This dual-method approach aims to enhance both detection and spread simulation, providing a dynamic and comprehensive model for understanding and mitigating wildfire risks.

The core of this thesis consists of three main chapters:

1. Overview of Dynamic Systems: This chapter 1 provides an introduction to dynamic systems, including definitions, terminology, and types of dynamic systems. It sets the foundation for understanding the concepts discussed in the subsequent chapters.

2. Dynamic Systems Prediction (DSP) Techniques Literature Review: This chapter 2 provides a comprehensive review of techniques used in Dynamic Systems Prediction (DSP). It explores existing literature to analyze various methods employed in predicting their behavior. Additionally, this chapter presents important research on Forest Fire Prediction and Monitoring.

3. Methodology and Implementation: This Last chapter 3 outlines the methodology adopted in the thesis, focusing on the selection and implementation of specific techniques for the proposed system. It details the methodology taken to achieve the research objectives and develop the intelligent system for predictive analytics.

Finally, this master thesis concludes with a general summary and offers perspectives on potential directions for future research.

Chapter 1

Overview of Dynamic Systems

1 Introduction

Dynamic systems, in a broad sense, refer to systems that evolve or change over time. They are characterized by the interplay of various components or variables that interact with each other, leading to dynamic behaviors or outcomes. These systems can be found across numerous disciplines, including physics, engineering, biology, economics, social sciences, etc.

In this chapter, the definition and characteristics of dynamic systems are presented, along with the fundamental concepts and terminologies essential for understanding how these systems operate. Different types of dynamic systems and their classifications are also explored, highlighting the key features that distinguish them from one another. Additionally, the characteristics and behaviors of dynamic systems are discussed, emphasizing their significance in various real-world applications.

2 Definitions

Dynamic systems are sets of interconnected components or variables that evolve or change over time, often exhibiting complex behaviors or patterns arising from the interactions among these components. They are characterized by their temporal dynamics, where the state of the system at any given time depends not only on its current state but also on its history and the influences of external factors [1] [2] [3] [4].

According to Ogata [5], a system is called dynamic if its present output depends on past input; if its current output depends only on current input, the system is known as static. The output of a static system remains constant if the input does not change. The output changes only when the input changes. In a dynamic system, the output changes with time if the system is not in a state of equilibrium.

3 Dynamic systems terminology

In dynamic systems, concepts such as state variables, dynamics, and equilibrium are pivotal, shaping our understanding across disciplines like mathematics and physics. These principles govern the behavior of interconnected components over time [5].

Within dynamic systems, a system's state at any moment is marked by a point in a state space, with its evolution determined by a specific function. This function acts as the system's rule for development, dictating future states based on the present one. In deterministic systems, a single future state is determined, while stochastic systems incorporate random events into the evolution of state variables [5] [6].

- **State variables** are quantities that uniquely define a system's state at any given time. Typically represented by real numbers or vectors in a geometric space, these variables pinpoint a specific location in the state space.
- **Dynamics** refer to the evolution rule governing a system, outlining the progression from the current state to future states. This function can be deterministic, leading to a single future state, or stochastic, introducing randomness.
- **Equilibrium** in dynamic systems signifies a state where the system remains unchanged over time, often achieved when forces within the system are balanced. Mathematically, an equilibrium point is where the derivative of the state variable is zero.

4 Types of Dynamic Systems

Dynamic systems can be classified into four main categories: (1) distributed versus "lumped" systems, (2) continuous-time versus discrete-time systems, (3) time-varying versus time-invariant systems, and (4) linear versus nonlinear systems [7].

4.1 Distributed versus Lumped Systems

In a distributed system, an infinite number of "internal" variables are required, leading to the system being governed by Partial Differential Equations (PDEs). On the other hand, a lumped system involves a finite number of "internal" variables, resulting in the system being governed by Ordinary Differential Equations (ODEs).

4.2 Continuous-Time versus Discrete-Time Systems

Continuous-time systems have variables and functions defined for all time, while discrete-time systems have variables defined only at discrete time points. Continuous-time systems are analog domain-based, involving variables like position , whereas discrete-time systems are digital domain-based, with variables like sampled position existing at discrete-time points.

4.3 Time-Varying versus Time-Invariant Systems

In a time-varying system, system parameters change over time (e.g., varying friction coefficient), whereas time-invariant systems have constant parameters. The variation of system parameters should not be confused with the variation of dynamic variables.

4.4 Linear versus Nonlinear Systems

Linear dynamic systems follow the principle of superposition, meaning the response caused by two or more stimuli is the sum of the responses that would have been caused by each stimulus individually. Nonlinear dynamic systems, on the other hand, do not adhere to this principle, making them more complex and often more representative of real-world systems [7].

5 Characteristics of dynamic systems

Dynamic systems characteristics define their behavior and functioning, with dynamic characteristics specifically focusing on criteria for instruments that change rapidly with time. These criteria include [8]:

- **Response Speed:** This characteristic denotes how quickly a system reacts to changes in its input or environment, with faster response times being preferable.
- **Fidelity:** Fidelity represents the accuracy with which a system reproduces its output in response to changes in input, indicating a faithful representation of the input.
- **Lag:** Lag refers to the undesirable delay between input changes and the system's response, highlighting the importance of minimizing lag for real-time applications.
- **Dynamic Error:** Dynamic error signifies the deviation between the desired output and the actual system response, with lower dynamic error being more favorable.

6 Predicting and monitoring Dynamic Systems

Dynamic systems, prevalent in natural and engineered domains, profoundly influence our world, spanning various disciplines from ecological balance to financial markets. Research in dynamic system prediction and monitoring primarily revolves around developing intelligent systems. These systems aim to forecast and monitor the behavior of dynamic systems, which evolve over time. The goal is to create models capable of accurately predicting future states and monitoring their real-time evolution.

a) **Predicting Dynamic Systems** The prediction aspect involves developing mathematical models and algorithms to anticipate the future states of a dynamic system based on its past and current states. This often requires the use of Artificial Intelligence (AI) techniques and historical data to make accurate predictions about future states [9].

b) **Monitoring Dynamic Systems**

On the other hand, monitoring involves the continuous observation of a dynamic system to track its state and performance over time. This can involve the use of sensors and other data collection devices to gather real-time data about the system. The collected data can then be analyzed to detect any significant changes or anomalies that could indicate a problem or a shift in the system's behavior [10] [11].

c) **Intelligent Systems** The term 'Intelligent Systems' refers to systems that use advanced computational techniques, to perform tasks that typically require human intelligence. In the context of dynamic system prediction and monitoring, 'Intelligent Systems' can learn from data, adapt to changes, and make informed decisions or predictions about future states. [12].

7 Dynamic system representations

7.1 Mathematical Tools for Modeling Dynamic Systems

Dynamic systems are modeled using various mathematical tools, each with specific applications and advantages [13]:

- **Differential Equations:** Differential equations are fundamental in modeling continuous dynamic systems where changes occur smoothly over time. For instance, they are used in physics to describe the motion of particles under forces, in biology to model population dynamics, and in engineering to design control systems. The power of differential equations lies in their ability to capture the continuous nature of most real-world processes, allowing for precise and detailed analysis of system behavior.
- **Difference Equations:** Difference equations are ideal for systems that evolve in discrete steps rather than continuously. Examples include financial models that predict stock prices at the end of each day or population models where generations do not overlap. They are simpler to solve compared to differential equations and are particularly useful in computer simulations where time is often discretized. Difference equations are widely used in digital signal processing and in the design of algorithms for predictive control in discrete-time systems.
- **State-Space Representations:** State-space models provide a comprehensive framework for representing complex dynamic systems. By expressing the system's state as a vector and its evolution as a set of linear or nonlinear equations, these models can handle multiple interdependent variables simultaneously. This approach is highly beneficial in control theory, robotics, and aerospace engineering, where it is crucial to monitor and control various aspects of a system's state. State-space representations also facilitate the application of modern control techniques like state feedback and optimal control.

7.2 Artificial Intelligence (AI) Tools for Modeling Dynamic Systems

AI tools offer advanced capabilities for modeling dynamic systems, leveraging data-driven approaches to enhance prediction and control:

- **Machine Learning (ML):** ML algorithms are trained on historical data to recognize patterns and make predictions about future states. For example, in predictive maintenance, ML models can forecast equipment failures based on historical usage and sensor data, reducing downtime and maintenance costs. In environmental modeling, ML can predict changes in weather patterns or pollutant levels. The flexibility of ML allows it to be applied across diverse fields, from healthcare to finance, improving decision-making through data-driven insights [14].

- **Deep Learning (DL):** DL, a subset of ML, involves neural networks with multiple layers that can model complex relationships within data. In dynamic systems, DL is used for tasks requiring high-dimensional data analysis, such as image-based object tracking in autonomous vehicles, natural language processing for real-time translation systems, and speech recognition for interactive voice response systems. DL's ability to handle large datasets and uncover intricate patterns makes it suitable for applications where traditional modeling techniques fall short [15].

- **Reinforcement Learning (RL):** RL algorithms learn to make optimal decisions by interacting with the environment and receiving feedback in the form of rewards or penalties. This approach is particularly effective in scenarios where the system's dynamics are complex and not fully known. For instance, RL can be used to develop adaptive control strategies for robotics, optimize resource allocation in network management, or design trading algorithms in financial markets. By continually learning from the environment, RL systems can adapt to changing conditions and improve their performance over time [16].

- **Agent-based modeling and simulation (ABMS):** ABMS has emerged as a powerful tool for studying complex systems characterized by the interaction of autonomous agents [17]. Unlike traditional modeling approaches that rely on aggregate-level equations or rules, ABMS focuses on representing individual agents and their interactions within an environment. This allows for the exploration of emergent phenomena that arise from the interactions of agents, making ABMS particularly well-suited for modeling dynamic systems with nonlinear and unpredictable behaviors [18]. ABMS has found applications across diverse domains, including transportation, urban planning, epidemiology, ecology, and economics. It offers a comprehensive approach to understanding and predicting behavior of dynamic systems by capturing the interactions between agents at a micro-level, which leads to the emergence of macro-level patterns and behaviors [17].

  At the core of ABMS is the concept of an agent, an autonomous entity capable of perceiving its environment, making decisions, and interacting with other agents. Agents exhibit a wide range of behaviors influenced by their internal state, environmental information, and interactions with other agents. The modeling of environments provides the context for agent interactions and influences agent behavior through feedback mechanisms. Interactions between agents give rise to emergent phenomena such as self-organization, cooperation, and competition, which are essential for understanding system dynamics [19]. Methodologies of ABMS include behavior-based modeling, rule-based modeling, and cellular automata [20] [21].

8 AI Methodology for Capturing System Dynamics

AI models capture the dynamics of various systems by learning patterns and relationships from data to make predictions or decisions. This process is as follows [22]:

- **Data Collection.**

- **Feature Extraction.**
- **Model Selection.**
- **Training and validation.**

9 Application areas

Dynamic systems theory offers a powerful framework for addressing real-world problems and challenges across a wide range of disciplines. Applications utilizing dynamic systems theory to model real-world phenomena encompass:

a) **Control Systems**

Dynamic systems theory is fundamental in control systems engineering, where it is applied to design and analyze feedback control systems. These systems play a critical role in regulating the behavior of dynamic processes across industries such as aerospace, automotive, manufacturing, and robotics [23].

b) **Economics and Finance**

Dynamic systems models find application in economics and finance to examine economic phenomena encompassing market dynamics, business cycles, and investment behavior. These models are utilized in diverse areas including macroeconomic modeling, financial market analysis, and forecasting [24] [25] [26].

c) **Ecology and Environmental Science**

Dynamic systems theory is pivotal in understanding ecological systems, addressing population dynamics, ecosystem interactions, and biodiversity. It enables ecologists to model and analyze intricate ecological processes and anticipate the consequences of environmental changes [27].

d) **Epidemiology**

Dynamic systems models are indispensable in epidemiology, examining the spread of infectious diseases like COVID-19, influenza, and HIV/AIDS. They enable public health officials to assess disease transmission dynamics, evaluate intervention strategies, and formulate informed policy decisions [28].

e) **Sociology and Psychology**

Employing dynamic systems approaches, researchers in sociology and psychology investigate social and psychological phenomena, including group dynamics, cultural evolution, and individual behavior. These approaches facilitate the understanding of how social systems evolve over time and the interactions among individuals within them [29].

f) **Mechanical Engineering**

Dynamic systems theory is extensively utilized in mechanical engineering to model and analyze mechanical systems such as engines, vehicles, and machinery. It assists engineers in predicting system behavior, optimizing performance, and designing control systems for stability and reliability [30].

g) **Neuroscience and Brain Dynamics**

Dynamic systems models find application in neuroscience for investigating the dynamics of neural circuits and brain function. They contribute to understanding how neural networks process information, generate behavior, and adapt to changing environments [31].

h) **Climate Science**

Utilizing dynamic systems theory, climate science models and simulates Earth's climate system, encompassing atmospheric circulation, ocean currents, and climate feedback mechanisms. These models assist in climate prediction, impact assessment, and policy formulation [32].

10 Conclusion

This chapter provided an overview of dynamic systems, emphasizing their evolving nature and broad applications across disciplines. Key concepts like state variables, dynamics, and equilibrium were discussed, along with the classification of dynamic systems into various categories. The importance of studying dynamic systems for prediction and monitoring purposes, as well as their applications in fields such as control systems, economics, and ecology, was highlighted.
The next chapter will delve into Dynamic Systems Prediction (DSP) techniques through a comprehensive literature review, with a particular emphasis on modeling and predicting forest fires as a case study. This exploration will encompass various methodologies used to forecast fire behavior, emphasizing advancements and practical applications in the field.

Chapter 2

Dynamic Systems Prediction (DSP) Techniques Literature Review

1 Introduction

Accurately forecasting the behavior of dynamic systems remains a significant challenge across diverse scientific disciplines. Characterized by constant evolution and intricate interdependencies between constituent parts, these systems often defy straightforward modeling using conventional techniques. However, recent breakthroughs in Artificial Intelligence (AI) offer promising avenues to address this longstanding predicament. In this chapter, the potential of AI techniques for predicting dynamical systems is analyzed, highlighting its advantages and limitations. Subsequently, an in-depth analysis of the Cellular Automata (CA) model is provided, detailing its structure, rules, and effectiveness in DSP. Finally, a case study on the application of AI to forest fire prediction and monitoring is presented, illustrating the practical implications of these techniques.

2 Overview of AI Techniques in DSP

Predictive analytics has been revolutionized by intelligent systems infused with AI, enabling to glean valuable insights from vast data pools and accurately foresee future outcomes. This section presents popular techniques for predicting the dynamic behaviors of systems.

2.1 Machine learning

Machine learning (ML) is a subfield of AI that enables computers to learn and improve from experience without being explicitly programmed. By analyzing data, ML algorithms identify patterns, make predictions, and refine their accuracy with more information. This capability supports a wide range of applications, from recommending movies and detecting fraud to assisting in medical diagnoses and enabling self-driving cars, significantly impacting various fields [33].

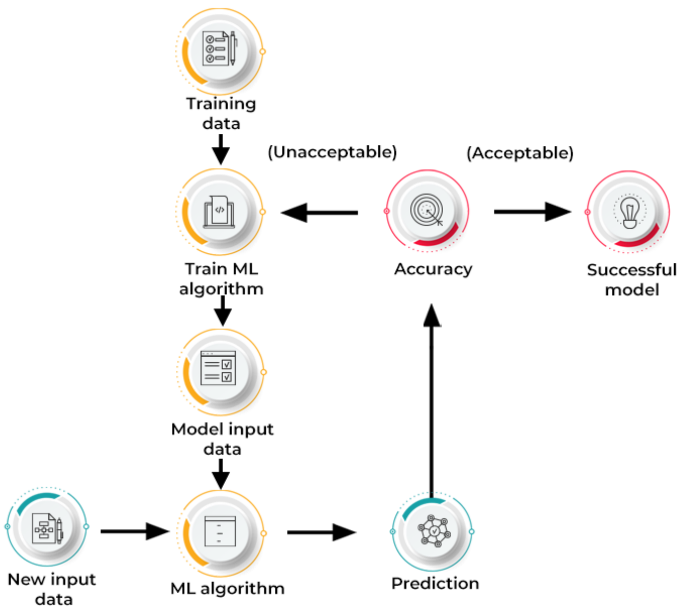

Figure 2.1: Machine learning prediction process [W1].

According to Figure 2.1, the machine learning process begins with inputting training data into a chosen algorithm. This training data can be known or unknown and influences the algorithm's development. New input data is then used to test the algorithm's accuracy. If predictions do not match the actual results, the algorithm is retrained repeatedly until it achieves the desired accuracy. This iterative process allows the algorithm to continually improve and provide increasingly accurate predictions over time.

Key ML techniques, such as Decision Trees (DT), Random Forest (RF) and Support Vector Machine (SVM) are extensively applied in dynamic system prediction. The subsequent sections will provide brief descriptions of these most common machine learning systems [34] [35] [36].

2.1.1 Decision Tree (DT)

A decision tree is a hierarchical technique where each path from the root represents a sequence of data separations, leading to a boolean outcome at the final node [34]. This structure, consisting of nodes and connections, simplifies complex decision-making processes. The primary advantage of decision trees is their ability to transform intricate problems into straightforward, visual processes, making the solutions more understandable and interpretable [37].

A DT consists of several key components:

- **Root Node:** The base of the DT.
- **Decision Node:** A sub-node that splits into additional sub-nodes.
- **Leaf Node:** A sub-node that does not split further, representing possible outcomes.
- **Sub-tree(Branch):** A subsection of the DT composed of multiple nodes.

Figure 2.2 illustrates the structure of a decision tree.

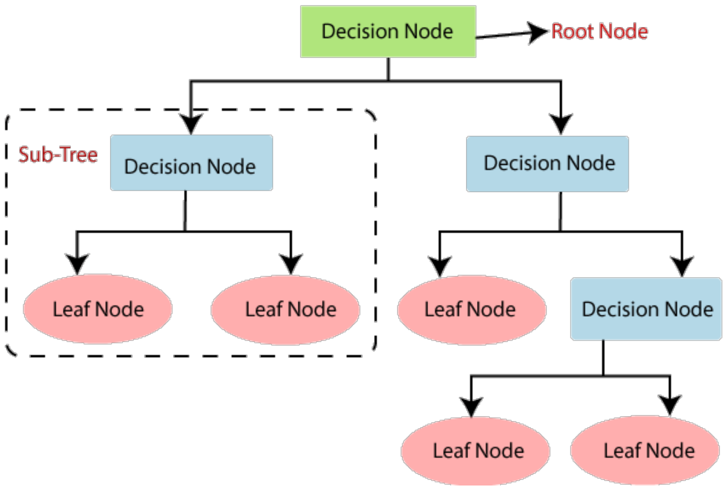

Figure 2.2: Decision tree [W2].

DT models are widely used in Electric Power dynamic Systems research for predicting damping ratios of inter-area oscillations and enhancing small-signal stability through generation rescheduling [38] [39]. Additionally, in healthcare informatics, DTs play a significant role in various dynamic systems, such as predicting diseases, identifying risk factors, and aiding in clinical decision-making, particularly in the context of cardiovascular disease prediction [40] [41].

2.1.2 Random Forest (RF)

Random Forest is an algorithm based on DTs. Unlike a single DT, Random Forest constructs multiple trees to make predictions, which helps reduce overfitting (figure 2.3).

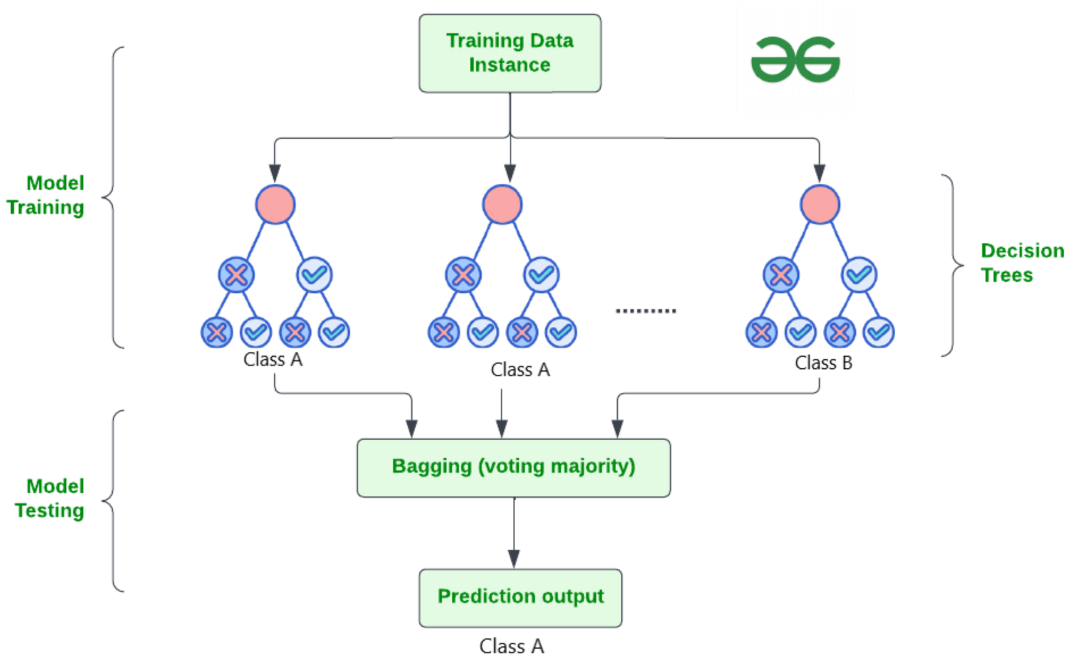

Figure 2.3: Random forest [W3].

This technique, known as bagging, involves using multiple trees to classify a feature vector, with individual classifications aggregated to determine the final output. This approach enhances prediction accuracy and generalization [42].
The Random Forest (RF) model has gained widespread application across various dynamic systems due to its adaptability and robustness. In healthcare, RF is employed for early diagnosis and prognosis, enhancing patient outcomes [43]. In finance and banking [44], it predicts mortgage defaults and detects fraudulent activities [45] [35], ensuring financial stability [36]. The model is also utilized in the stock market to forecast price fluctuations, aiding investors in making informed decisions. Notably, during the COVID-19 pandemic [46], RF played a crucial role in guiding public health actions by simulating virus spread [47] and assisting authorities in decision-making [48].

2.1.3 Support Vector Machine (SVM)

SVM is another significant model in machine learning, renowned for its effectiveness in high-dimensional spaces and its versatility in handling both linear and non-linear data. SVM operates by finding a hyperplane in an N-dimensional space that distinctly classifies the data points. Through the process of maximizing the margin between the nearest data points in each class (Figure 2.4), SVMs identify the best hyperplane in an N-dimensional space to divide two classes of data points. Despite the complexity of this task, SVMs are known for their effectiveness and robustness [49].

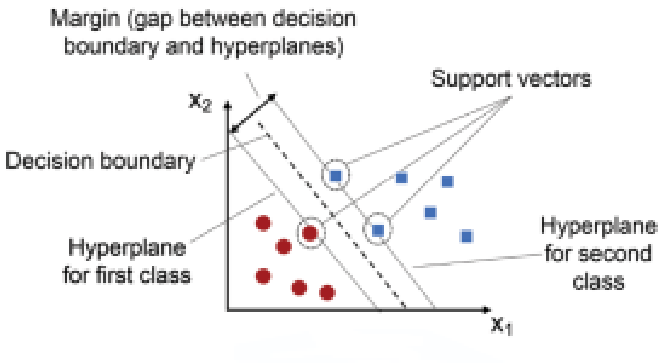

Figure 2.4: Linear SVM model [W4].

SVMs have found applications in various fields. They have been used for image-based analysis and classification tasks [50]. In the realm of geospatial data, SVMs have been used to handle noisy data and solve inversion problems [51] [52]. During the COVID-19 pandemic, SVMs were integrated with other models to simulate the spread of the virus, aiding authorities in making informed decisions [53] [54].

2.2 Deep learning

Deep learning, a subset of machine learning, empowers artificial intelligence systems to learn from data through artificial neural networks [55]. This technique has been extensively explored for many dynamic systems, including the prediction and detection of dynamic forest fires phenomena [56]. To leverage the potential of deep learning, numerous studies have been conducted on fire incidence modeling in this domain [57].

2.2.1 Deep Neural Network (DNN)

A DNN is an artificial neural network with multiple layers between the input and output layers (Figure 2.5). Each layer learns to transform its input data into a slightly more abstract and composite representation, allowing the network to model complex non-linear relationships, and generate compositional models where the object is expressed as a layered composition of primitives [58].

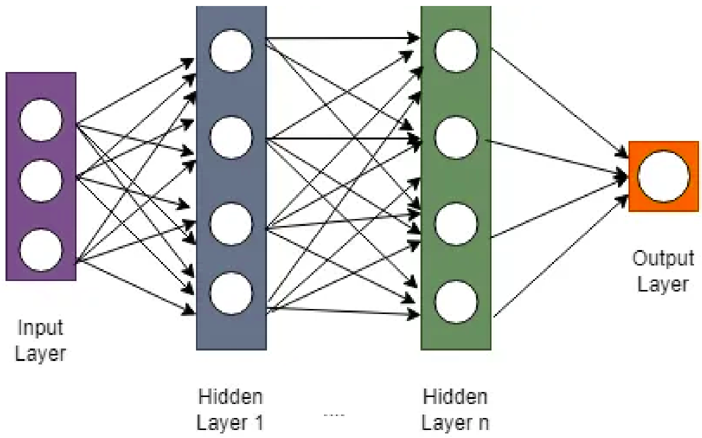

Figure 2.5: Deep neural network (DNN) [W5].

DNN model is used in many fields, including modeling of dynamical systems [12], solving flows of dynamical systems [59], integrating model-based and data-driven learning for dynamical system evolution [60], predicting dynamic thermal nonlinear processes [61] and demonstrating high accuracy in predicting short-term electricity consumption in dynamic power systems [62] [63].

2.2.2 Long Short-Term Memory (LSTM) networks

LSTMs were specifically designed to address the challenge of learning long-term dependencies in sequences. Unlike conventional Recurrent Neural Networks (RNNs), which struggle with capturing long-term relationships, LSTMs possess a unique architecture that allows them to retain and utilize information over extended periods. This capability is due to the presence of "cells" with memory units and gates that regulate information flow, enabling LSTMs to decide when to remember or forget information (Figure 2.6). Their exceptional memory retention makes LSTMs particularly effective for complex sequence tasks like machine translation and natural language processing [64].

LSTM have proven effective in predicting traffic patterns in urban areas, offering valuable insights for dynamic traffic management and congestion reduction [65] [66].

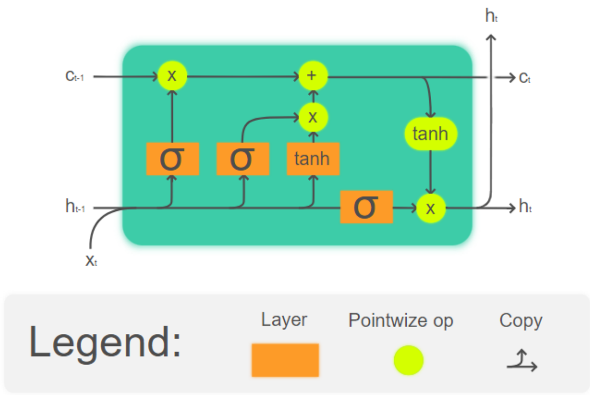

Figure 2.6: Long short-term memory (LSTM) [W6].

2.2.3 Autoencoder

Autoencoder is an unsupervised learning model, which can automatically learn data features from a large number of samples and can act as a dimensionality reduction method. Autoencoder consists of two parts: Encoder and Decoder [67] (Figure 2.7).

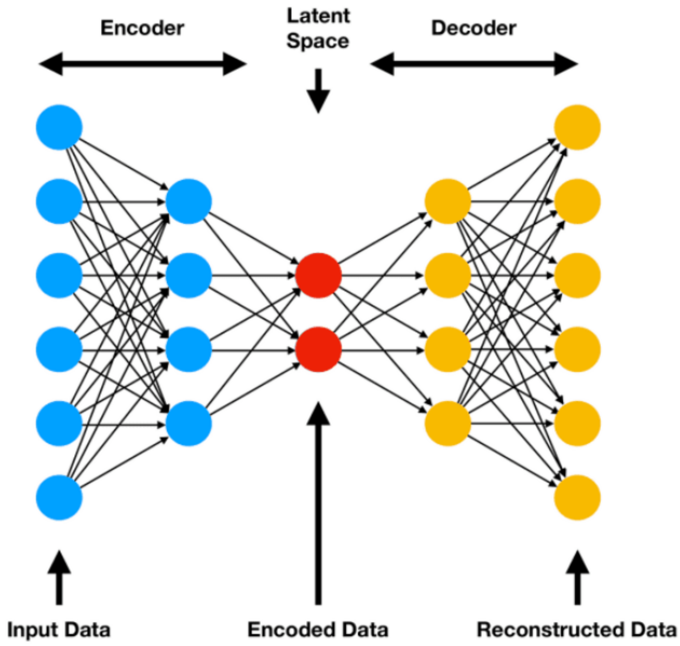

Figure 2.7: Autoencoder schema [W7].

Autoencoders are often applied for anomaly detection [68], data denoising [69], and dimensionality reduction. In dynamic manufacturing systems [70], autoencoders are utilized to detect anomalies in machine operations, thereby preventing equipment failures and ensuring continuous production flow [71].

2.3 Strengths and Limitations of ML and DL in DSP

AI has emerged as a powerful tool in predicting the behavior of dynamic systems, offering both strengths and limitations that warrant discussion AI has emerged as a powerful tool for predicting the behavior of dynamic systems, offering notable strengths as well as certain limitations [72]. This section will present both aspects:.

2.3.1 Strengths

- **Pattern Recognition:** AI techniques, including machine learning and deep learning, excel at recognizing complex patterns within large datasets.
- **Integration with Domain Knowledge:** AI techniques can be combined with domain knowledge to enhance prediction accuracy.

2.3.2 Limitations

- **Data Dependency:** The effectiveness of AI for dynamic system prediction relies heavily on the availability and quality of data. In situations where data is sparse or unreliable, AI models may struggle to make accurate predictions.
- **Generalization:** Robust generalization is a key challenge in AI-based prediction of dynamic systems..
- **Ethical and Societal Implications:** The use of AI in predicting dynamic system behaviors raises ethical and societal concerns, particularly regarding privacy and the potential for unintended consequences.

Given these limitations, Cellular Automata (CA) model, provides a robust framework for simulating complex system behaviors through simple, discrete state changes.

2.4 Overview of Cellular Automata (CA) in DSP

Cellular Automata (CA) have emerged as a powerful paradigm for simulating and predicting the behavior of dynamic systems. This section provides an in-depth analysis of the structure, rules, and effectiveness of CA in simulating complex dynamic systems.

2.4.1 CA Concept

Cellular Automata (CA) are discrete computational systems that consist of a grid of cells, each of which can be in one of a finite number of states. The state of each cell evolves over discrete time steps according to a set of rules based on the states of its neighboring cells. Cellular automata are used to model complex systems and phenomena by simulating the behavior of each cell and observing the patterns and structures that emerge from their interactions [73].

2.4.2 CA Structure

Typically, it consists of a regular grid of cells interconnected with their neighbors, with each cell possessing a finite set of states. While the grid can exist in various dimensions, in this work, we focus on the two-dimensional grid structure [74].

1. **The Cell**

   The fundamental unit of CA is the cell, which functions as a Finite State Automaton evolving according to predefined update rules. Each cell's state is influenced by its previous states and the states of its neighboring cells. While cells typically form a lattice structure, there are no inherent limitations, and alternative configurations are permissible. Furthermore, cells need not be identical, allowing for diversity within the grid [74].

2. **Update Rules**

   The state of a CA is determined by a set of update rules governing the transition from the current state to the next state for each cell. These rules are predefined and dictate how each cell's state evolves based on its current state and the states of its neighbors. The update process is iterative, with each iteration representing a discrete time step in the simulation [74].

3. **Interaction with Neighbors**

   Central to the operation of CA is the interaction between cells and their neighbors. Various methods exist for defining cell neighborhoods, with the most common being the 4-connected Von Neumann neighborhood and the 8-connected Moore neighborhood. The choice of neighborhood configuration influences the extent of interaction and spatial relationships within the model [74]. CA are utilized across a wide range of fields, demonstrating their versatility in simulating complex dynamic systems. They are used in the evolutionary dynamics of social networks to simulate and analyze network evolution [75]. In spatial modeling, CA employ the Voronoi spatial model to manage neighborhood relations and generate complex global patterns.

   In urban modeling [76]. CA simulate dynamic states on a local scale to aid in urban development planning [77]. They are also applied in road traffic simulation to analyze traffic flow dynamics [78]. In biology, CA model complex biological phenomena [79], while in epidemiology, they have been employed to model the Covid-19 pandemic [80], showcasing their wide-ranging applications in public health and other domains.

2.5 A Case Study: Forest Fires

2.5.1 Phenomenon of a forest fire

The phenomenon of a forest fire can be introduced as a dynamic system due to its complex and evolving nature over time [81] [82] [83].

2.5.2 AI-based Research on Forest Fire Prediction and Monitoring

a) **Machine learning-based systems** In the realm of forest fire prediction, machine learning techniques such as the DT algorithm, along with its ensembles, have emerged as valuable tools. DT, with its ability to predict, explain, and classify outcomes within a tree structure, offers a transparent and interpretable model. Stojanova's research [84] exemplifies the effectiveness of DT and its ensembles in predicting forest fires.
   This body of evidence not only underscores the importance of DT and its ensembles

in forest fire prediction but also highlights their significance in evaluating associated risks, further emphasizing their value in this critical field of study.

In addition, research conducted by various authors [85] [86] has consistently highlighted the superior predictive capabilities of RF in anticipating forest fires [87].

Furthermore, the Support Vector Machine (SVM) model has garnered attention as a robust predictive tool for forest fire damage. By maximizing the margin between nearest data points in each class, SVM identifies the optimal hyperplane in an N-dimensional space, effectively distinguishing between different classes of data points. Numerous studies have underscored SVM's effectiveness in this domain. For example, Xie [88] leveraged SVM to analyze historical data and environmental factors, while Bayat [89] found SVM to possess the highest predictive ability in forecasting burned area sizes. Cortez P. [90], meanwhile, applied SVM alongside weather inputs to predict the burned areas of small fires. Collectively, these studies highlight SVM's potential to enhance forest fire management strategies through its capacity to analyze complex datasets and provide accurate predictions.

b) **Deep Learning-based system**

Recent advancements in forest fire modeling using deep learning (DL) have shown significant progress in simulation techniques. For instance, Cheng-Yu et al. [91] developed a DNN model that utilized demographic, architectural, and economic data to forecast fire incidents. The model demonstrated remarkable performance metrics closely aligned with ideal benchmarks, highlighting its efficacy. Similarly, Lai et al. [92] enhanced prediction accuracy through a DNN system analyzing meteorological data, geographical information, and historical fire records, effectively addressing imbalanced data challenges. Additionally, Sam-Keun Jae-Geun [93] showcased the potential of DNN models in predicting the burned area of forest fires, particularly small-scale incidents, utilizing meteorological data.

The Long Short-Term Memory (LSTM) model has emerged as a pivotal tool in forest fire prediction, as evidenced by a study [94] focusing on wildfire scale prediction. This study leveraged meteorological data and fire records, where LSTM outperformed other neural network models such as Back Propagation Neural Network (BPNN) and Recurrent Neural Network (RNN), particularly in early-stage wildfire prediction [95].

Furthermore, the utilization of autoencoders represents a significant advancement in forest fire prediction. Can Lai et al. [96] highlighted an autoencoder-based DNN technique to address challenges posed by imbalanced data distributions in forest fire datasets. By leveraging autoencoders, the DNN model overcomes these hurdles, enhancing prediction accuracy, and emphasizing the importance of advanced machine learning techniques in augmenting wildfire management and prevention endeavors [97].

c) **Cellular Automata-based systems** Recent work on forest fire modeling using cellular automata (CA) has shown significant advancements in simulation techniques such as: Aleixo et al. [98] employed a CA framework to simulate local fire dynamics, identifying phase transitions for various fire risk combinations and using these values to parameterize the landscape network. Hui and Rui [99] proposed a simulation algorithm integrating a geographic CA to address the high error rates and inefficiencies of traditional models in large-scale forest fire simulations. Mahdizadeh and Navid [100] developed a CA model to simulate wildfire spread, accounting for key spatial and temporal factors such as wind speed and direction, vegetation type and density, and topographical conditions. Jellouli et al.[101] investigated the use of CA methods for forest fire simulation, incorporating parameters like natural vegetation, density, humidity, wind force, and elevation. In addition, a forest fire prediction model developed by Xuan Sun and colleagues incorporates other CA models such as the Wang Zhengfei model to simulate the spread of fire based on neighboring cell interactions. By combining CA principles with Machine Learning techniques, the model improves the precision of forecasting forest fire propagation, providing valuable insights for effective risk management and firefighting strategies [102] [103].

The strengths and Limitation of different methodologies discussed in the preceding literature on forest fire prediction, are outlined in articles [104] [105] [56] [106] [107].

The selection of methods was informed by their respective advantages. Therefore, Deep Neural Networks (DNN) and Cellular Automata (CA) were chosen to complement each other's inherent disadvantages, thereby leveraging the strengths of both technologies to achieve optimal performance.

3 Conclusion

This chapter delved into the role of intelligent systems, particularly AI techniques, in predictive analytics, focusing on their applications in forecasting dynamic system behaviors. It explored various AI methods such as CA, ML, and DL, highlighting their strengths, limitations, and real-world applications. While CA offers insights into complex system behaviors through local interactions, ML techniques like RF and SVM prove effective in predicting forest fires despite challenges with data size and computational efficiency. ML models, such as DNNs and LSTM, further enhance prediction accuracy, especially in handling imbalanced data and time-series forecasting. Overall, the chapter emphasized the transformative potential of intelligent systems in predictive analytics, advocating for continued research and development in the field. Based on the analysis of previous AI methods, two techniques have been selected for the proposed system, which are presented in Chapter 3.

Chapter 3

Methodology and Implementation

1 Introduction

The escalating threat of forest fires to ecosystems, wildlife, and human life necessitates the development of advanced systems for early detection, monitoring, and prediction. With climate change driving an increase in both the frequency and intensity of these fires, the need for innovative solutions leveraging modern technology has never been more urgent. This chapter details the design and methodology of a proposed system aimed at addressing these challenges through advanced data processing, Deep Learning (DL) prediction model, and Cellular Automata (CA) for modeling fire spread.

2 Proposed System: FireTrack

Existing forest fire detection systems, as discussed in the previous chapter, tend to emphasize either detecting fires in specific areas or tracking their spread. However, addressing only one of these aspects falls short of providing a comprehensive fire management strategy.
Therefore, this work proposes an advanced system, FireTrack, that integrates both a specific Deep Learning technique and a CA model for predicting and monitoring forest fires. This system is designed to enhance detection capabilities while simultaneously tracking the progression of fires. By merging these technologies, the proposed solution aims to offer a more robust and complete approach to forest fire management.

3 Architecture of the proposed FireTrack system

The FireTrack system architecture leverages a specific DNN architecture for predicting fire ignition points [108]. It then utilizes a CA model to simulate fire spread. The following figure illustrates the overall architecture of the proposed system. Each subsystem will be described in detail in the following sections.

According to Figure 3.1, the proposed system is divided into three subsystems:

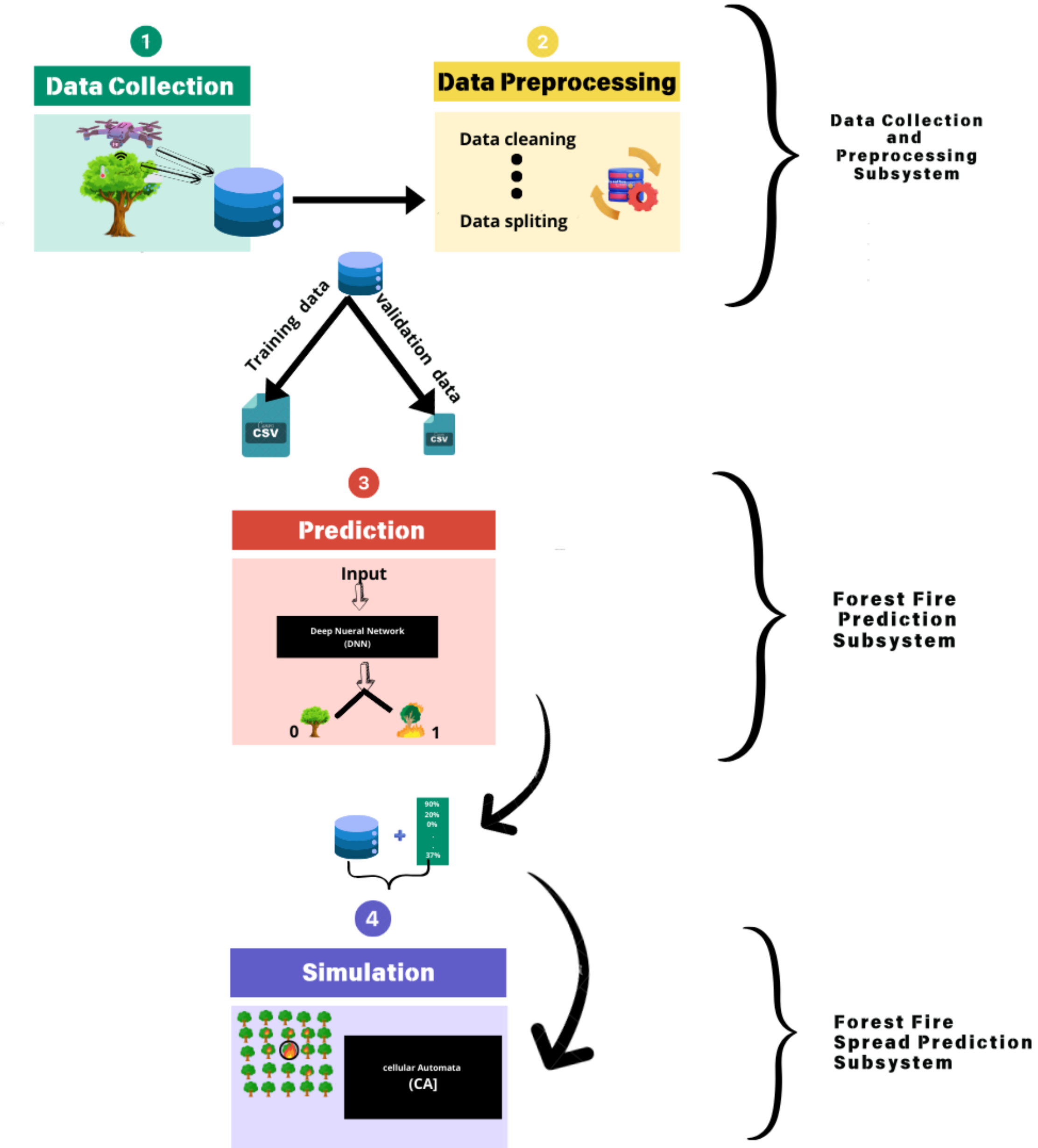

Figure 3.1: Proposed system architecture.

3.1 Data Collection and Preprocessing Subsystem

a) Data Collection

In this step, the focus is on gathering meteorological data such as temperature, humidity, wind speed, and wind direction from sensors or with drones. However, due to the unavailability of sensors and drones, in this work, a meteorological dataset is used as a substitute to fulfill the role of these data collection mechanism.

The primary used dataset comes from Algeria and consists of 244 cases with meteorological observations from the northeastern Bejaia region and the northwestern Sidi Belabbas region, recorded during June to September 2012 [W13]. This period was chosen due to the high incidence of fires recorded from 2007 to 2018.

The dataset includes temperature, relative humidity, and wind speed to predict fire and non-fire occurrences, with 138 instances classified as "fire" and 106 as "not fire." (Figure 3.2).

```
df = pd.read_csv('Algerian_forest_fires_dataset.csv')
df.columns
df.head()
```

	day	month	year	Temperature	RH	Ws	Rain	FFMC	DMC	DC	ISI	BUI	FWI	Classes	Region
0	1	6	2012	29	57	18	0.0	65.7	3.4	7.6	1.3	3.4	0.5	not fire	1
1	2	6	2012	29	61	13	1.3	64.4	4.1	7.6	1.0	3.9	0.4	not fire	1
2	3	6	2012	26	82	22	13.1	47.1	2.5	7.1	0.3	2.7	0.1	not fire	1
3	4	6	2012	25	89	13	2.5	28.6	1.3	6.9	0.0	1.7	0.0	not fire	1
4	5	6	2012	27	77	16	0.0	64.8	3.0	14.2	1.2	3.9	0.5	not fire	1

Figure 3.2: Algerian forest fires dataset.

b) **Data Preprocessing**

Data preprocessing is a vital step for ensuring data quality and prediction accuracy. This process involves several key activities: deleting duplicate rows, removing erroneous data, handling missing data, and performing feature selection to enhance the dataset's overall quality.

Duplicate data, often resulting from errors or merging datasets, can distort analytical results and is identified through exact or partial matching and subsequently removed. Erroneous data, which includes incorrect or inconsistent data points, is detected using exploratory data analysis and can be corrected or removed to ensure dataset accuracy and reliability. Mean imputation, a technique for handling missing data, replaces missing values with the mean of the observed data for that variable, thereby maintaining the overall distribution and central tendency of the data. Mathematically, if $(x_1, x_2, \ldots, x_n)$ are the observed values of a variable with missing data, the mean $(\bar{x})$ is calculated as:

$$\bar{x} = \frac{1}{N}\sum_{i=1}^{n} x_i \tag{3.1}$$

where n is the number of observed values. Each missing value is then replaced with $\bar{x}$ .This approach preserves consistency and sample size, particularly in datasets with a small proportion of missing values.

In this preprocessing step, the Algerian forest fires dataset undergoes a rigorous quality check. This ensures the data's integrity and suitability for the subsequent analysis.

Data satdardisation and Splitting In this study, an 80%-20% train-test split was initially employed to allocate data for model evaluation. Various ratios, including 70%-30%, 60%-40%, 80%-20%, and 50%-50%, were tested to determine the optimal split, as supported by scientific literature. Xu and Goodacre [109] [110] emphasize the importance of selecting an appropriate split for reliable evaluation, while Raschka [111] highlights the need to experiment with different splits to achieve optimal performance. After thorough testing, it was found that the 80%-20% split provided the best balance between training data sufficiency and testing data reliability, making it the optimal choice for this model (see Figure 3.3).

```
# separate dataset into train and test
X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.20,random_state=0)
X_train.shape, X_test.shape
```

Figure 3.3: Data splitting code.

3.2 Forest Fire Prediction Subsystem

This subsystem employs a Deep Neural Network (DNN) model. DNNs are well suited for this task due to their ability to capture complex patterns and relationships within the data compared to other techniques tested in this system During our model selection process, we conducted an extensive evaluation of various machine learning and deep learning techniques to identify the most effective approach. Based on this comprehensive analysis, we determined that the Deep Neural Network (DNN) model was the optimal choice due to its superior performance metrics. DNN model ensures high accuracy in prediction by leveraging multiple layers of nonlinear .The model takes the preprocessed meteorological data as input and outputs predictions regarding the likelihood of forest fires. processing units, which can effectively learn from the intricate features of the dataset. Feature selection is a crucial step in training a DL model that aims to improve model performance by reducing the dimensionality of the data. It involves selecting the most relevant features that contribute significantly to the prediction task, while eliminating redundant or highly correlated features.

Feature selection is a crucial step in training a DL model that aims to improve model performance by reducing the dimensionality of the data. It involves selecting the most relevant features that contribute significantly to the prediction task, while eliminating redundant or highly correlated features.

```
def correlation(dataset, threshold):
    col_corr = set()
    corr_matrix = dataset.corr()
    for i in range(len(corr_matrix.columns)):
        for j in range(i):
            if abs(corr_matrix.iloc[i, j]) > threshold:
                colname = corr_matrix.columns[i]
                col_corr.add(colname)
    return col_corr
corr_features = correlation(X_train, 0.75)
# features which has correlation more than 0.75
corr_features
```

```
{'BUI', 'DC', 'FWI'}
```

Figure 3.4: Code feature selection

Additionally, we exclude the date-related attributes (Day, Month, Year) and Region, as our focus is on environmental variables that directly influence the target variables [112].

The proposed DNN architecture starts with an input layer containing neurons, each representing a selected environmental variable. A dropout layer follows to prevent overfitting. The number of hidden layers and neurons was empirically selected. The output layer consists of a single neuron producing a probability value for forest fire occurrence. This proposed architecture balances complexity and efficiency, making it well-suited for accurate forest fire prediction. This approach enhances the model's accuracy underscoring its potential to significantly contribute to forest fire prediction and prevention efforts.

3.3 Forest Fire Spread Prediction Subystem

This subsystem predicts the spread of forest fires by employing a CA model. With previous prediction of fire points over the area, this system models the potential spread based on multiple factors such as altitude, temperature, wind, and others rules of fire spread. The integration of these predictions, with the environmental data from the previous system, allows for a comprehensive approach to simulating and monitoring the spread of forest fires, enhancing response strategies.

3.3.1 Proposed Configuration

To model the propagation of a forest fire using a CA model, we define a grid where each cell represents a small area of the forest. Each cell can be in one of three distinct states, represented by specific colors for visualization purposes (see Figure 3.5):

- **N (Non-burnt):** The cell, colored green, contains vegetation that is susceptible to catching fire.
- **B (Burning):** The cell, colored red, is currently on fire and actively spreading the fire to adjacent cells.

- **C (Burnt):** The cell, colored black, has been consumed by the fire and can no longer burn or spread the fire.

Cij	N	N	N	N
N	N	B	B	B
B	C	B	B	B
B	C	B	C	B
B	C	C	C	B

Figure 3.5: Example of Grid of cells

4 Fire Spread Rules

During my two-month internship at the Forest Conservation Service in Guelma (attached internship certificate), I gained comprehensive knowledge of the factors influencing the propagation of forest fires. This understanding was based on various environmental factors. The wind, including direction (North, South, East, West, and diagonal directions like North-East) and speed, plays a crucial role. Temperature indicates the ambient heat levels, while altitude affects oxygen levels and terrain, influencing how fires spread. These characteristics were used to create rules to model fire propagation behavior.
Understanding these factors is essential for predicting and managing the spread of forest fires effectively. The primary rules are that a burning cell provides each neighboring non-burning cell a base probability $\mathcal{P}_0$ of catching fire, ensuring a realistic simulation and the burning trees can generate sparks that travel to other parts of the forest, thereby spreading the fire further.

5 Environment

5.1 Hardware

The implementation of FireTrack system is performed on an AMD Quad-Core A8-7410 2.5GHz, RAM 8Gb.

5.2 Software And Programming language

- **Python(3.10):** is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together [113].

- **Netlogo(6.3.0):** is a multi-agent programmable modeling environment used for simulating natural and social phenomena. It is particularly well-suited for modeling complex systems that evolve over time. Researchers and educators use NetLogo to build models and run simulations to analyze how individual behaviors lead to collective outcomes in various scientific fields [114].

5.3 Used Libraries

- **Keras:** is a high-level Python library for building and training deep learning models. It provides an easy-to-use interface for quick prototyping and supports both convolutional and recurrent neural networks, running on top of TensorFlow, Theano, and CNTK backends. Keras offers various components for developing complex neural networks, such as layers, objectives, optimizers, and activation functions, along with tools for data preprocessing, model evaluation, and visualization. Its simplicity and user-friendliness make it a popular choice in both academia and industry, and its extensive documentation and active community further enhance its utility in developing deep learning applications [115].

- **Pandas:** is an open-source Python library designed to offer data structures for efficiently handling large and complex datasets. It provides a variety of tools for data cleaning, transformation, and exploration. Pandas is especially useful for managing tabular data, which is common in scientific applications such as social science and bioinformatics. It simplifies the process of working with structured data, making it a popular choice among researchers and data analysts [W14].

- **Scikit-learn (sklearn):** is a Python library that provides a wide array of machine learning algorithms for both supervised and unsupervised learning tasks. Built on top of established libraries such as NumPy, SciPy, and Matplotlib, it offers a user-friendly interface for data manipulation. Scikit-learn includes algorithms for classification, regression, clustering, and dimensionality reduction, as well as tools for data preprocessing, model selection, and evaluation. Its ease of use, flexibility, and scalability have made Scikit-learn popular in both academic research and industry. Additionally, its comprehensive documentation and active community of users and developers make it a valuable resource for those interested in machine learning [116].

- **NumPy:** is a Python library that facilitates numerical computing with a wide range of mathematical functions, providing robust and efficient support for large, multi-dimensional arrays and matrices. It is an essential tool for scientific computing in Python and is widely used in fields such as engineering, physics, and data science. NumPy's array objects are more efficient and powerful than Python's built-in data structures for numerical calculations. The library enables vectorized operations on arrays, significantly reducing computation time. Additionally, NumPy offers functionalities like linear algebra, Fourier analysis, random number generation, and tools for interfacing with other programming languages and libraries [117].

- **Matplotlib:** is a Python library for creating 2D plots, including static, animated, and interactive visualizations. It is widely used in scientific computing for data exploration and visualization. With a comprehensive set of 2D plotting functions and a high degree of customization, Matplotlib allows users to create sophisticated and complex plots with ease. It is compatible with various other Python libraries and frameworks, enhancing its versatility for visualizing data in different contexts [118].

- **PyQt5**: is a set of Python bindings for the Qt application framework, enabling the creation of cross-platform applications with rich graphical user interfaces, extensive widget libraries, and advanced features like database integration, web embedding, and multimedia handling [w15].

6 FireTrack interface

The FireTrack interface, as can be seen in Figure 3.6, consists of in the forest populated with pine and oak trees, represented by a grid of cells (or patches), where each cell may contain one or more trees, an empty space or a place burned by fire.

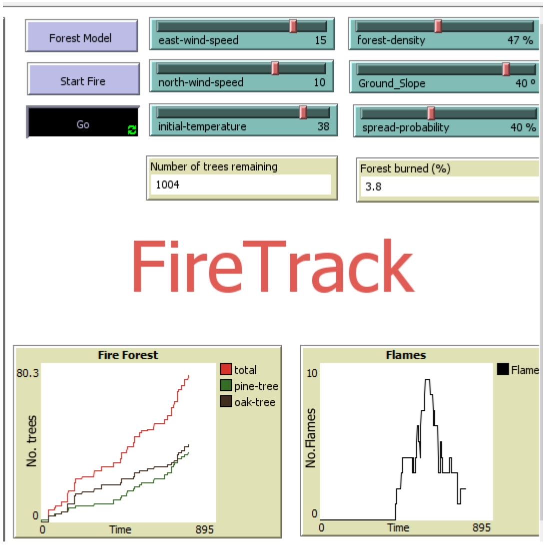

Figure 3.6: FireTrack System .

A set of global variables was defined, using the elements NetLogo graphics, to represent either properties of the forest itself or environmental factors that can affect the spread of fire. Among others, the following parameters:

- **Forest-density:** the density of the forest.
- **East-wind-speed:** the wind speed in the east direction .
- **North-wind-speed:** the wind speed in the north direction .
- **Initial-temperature:** the initial temperature for each cell.

7 Results and discussion

We will start with the evaluation matrices that we use to measure our model's performance.

7.1 Evaluation Metrics

The ability to measure a model's performance is critical for comparing various algorithms or models as well as for risk assessment. The primary purpose of performance metrics is to answer the issue of how accurate a model is at forecasting future events. In addition, a number of parameters must be calculated, such as:

- **True Positive (TP):** Both the actual and anticipated outputs were 1, as in the situations.
- **True Negative (TN):** The actual outcome was 0 compared to the instances' expected 0.
- **False Positive (FP):** When a case expected 1 but produced a 0.
- **False Negative (FN):** The output was 1 instead of 0, as the examples had projected.
- **Confusion Matrix:** The confusion matrix is a commonly used evaluation tool for classification tasks, suitable for both binary and multiclass classification problems. Table 3.1 shows the confusion matrix.

		Predicted classes	
		class = Negative	class = Positive
Actual classes	class = Negative	TN	FP
	class = Positive	FN	TP

Table 3.1: Confusion matrix.

- **Accuracy**

  Accuracy measures the proportion of correctly classified instances out of the total number of instances. It is calculated using the formula:

$$\textbf{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.2}$$

- **Recall**

  Recall, also known as sensitivity or true positive rate, determines the ratio of true positive predictions to all actual positive instances. It highlights the model's ability to identify positive instances. Recall can be calculated using the recall_score function and the following formula:

$$\textbf{Recall} = \frac{TP}{TP + FN} \tag{3.3}$$

- **Precision**

  Precision is another important metric that calculates the ratio of true positive predictions to all positive predictions. It specifically measures the accuracy of positive predictions. Precision can be derived using the precision_score function and the following formula:

$$\textbf{Recall} = \frac{TP}{TP + FP} \tag{3.4}$$

- **F1-score**

  The F1 score is a metric that combines precision and recall into a single value, providing a balanced measure of the model's performance. It takes into account both false positives and false negatives and is particularly useful in scenarios with imbalanced class distributions. The F1 score is the harmonic mean of recall and precision and can be computed using the f1_score function. It is calculated using the following formula:

$$\textbf{F1\_score} = 2 \times \frac{precision \times recall}{precision + recall} \tag{3.5}$$

7.2 Fire Forest prediction subsystem result

The performance of the proposed Deep Neural Network (DNN) model is notable for its robust predictive capabilities, as evidenced by various evaluation metrics. The model achieved an impressive accuracy of 0.98, demonstrating its ability to correctly classify the vast majority of instances. Precision and recall were recorded at 1 and 0.96, respectively, indicating that the model excels in identifying true positives while minimizing false negatives. Furthermore, the F1-score, which balances precision and recall, was also 0.98, underscoring the model's overall effectiveness in predictive performance. These results collectively highlight the DNN model's proficiency in accurately predicting forest fires, ensuring both high sensitivity and specificity in its predictions.

7.2.1 Classification Report

The classification report, as depicted in the Figure **??**, highlights the model's excellent overall performance, achieving an accuracy of 0.98. This high accuracy underscores the model's effectiveness in correctly classifying instances within the dataset. The detailed metrics in the report, including precision, recall, and F1-score, further affirm the model's exceptional performance across various evaluation criteria.

7.2.2 Precision Recall Curve

The ROC curve depicted in the Figure 3.7 closely approaches the ideal top-left corner, signifying a high true positive rate and a low false positive rate. This indicates the model's exceptional ability to accurately classify positive cases while minimizing the misclassification of negative cases. Such performance highlights the model's robustness and reliability in distinguishing between classes effectively.

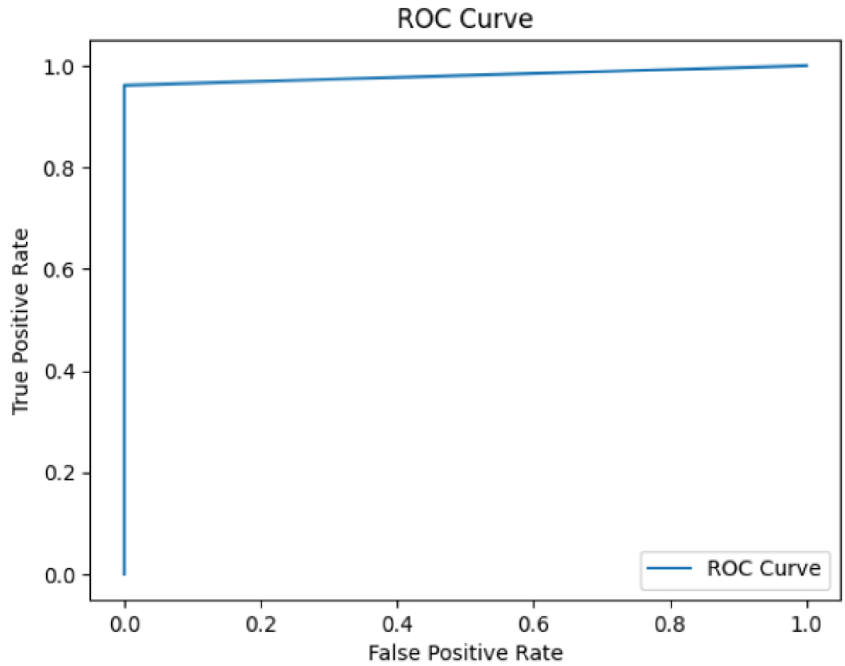

Figure 3.7: Precision Recall Curve.

7.3 Fire forest spread subsystem result

The results of a forest fire simulation are presented and analyzed across four distinct scenarios. These scenarios maintain similar model configurations, differing only in the parameters of wind speed in the north and east directions, and terrain slope. All relevant values are detailed in Table 3.2.

Properties	Scenario 1	Scenario 2	Scenario 3	Scenario 4
Initial Temperature	38°C	38°C	38°C	30°C
Inclination	30°	-30°	30°	-30°
North Wind	-10°	-10°	10°	10°
East Wind	15	15	15	15

Table 3.2: Configurations of different scenario.

The findings underscore the critical roles of wind conditions, and ignition patterns in forest fire dynamics, offering valuable insights for developing effective fire management strategies.

8 Conclusion

This chapter outlined the methodology and implementation of the proposed advanced system for forest fire management, leveraging modern technology tools for early detection, monitoring, and prediction. By detailing the system's architecture, data preparation, DNN structure, and proposed CA model, the robustness, and effectiveness of the system were demonstrated.

Evaluation metrics confirmed its reliability, and the integration of cellular automata simulation enhanced fire spread prediction by incorporating meteorological factors. Experimental results also validated the system's performance, showing significant improvements in forest fire management. The proposed approach underscores the crucial role of technology in mitigating the impacts of forest fires on ecosystems, wildlife, and human life.

Finally, the robustness of the proposed system was validated during my participation in the AI 24 Day event organized by the Computer Science Department at the University of Guelma, where I presented it as a workshop speaker (Certificate of participation attached in the end of this document). The model garnered positive feedback from the academic community, reinforcing its potential for practical, real-world application.

General conclusion

Forest fires are complex events influenced by numerous factors, including topography, vegetation, weather conditions, and ignition sources. Traditional fire management methods often fail to provide rapid and accurate predictions, leading to substantial ecological and economic losses. The main challenge is to develop a system that integrates these diverse factors to deliver reliable real-time predictions, thereby enhancing fire management effectiveness.

In this work, we proposed an intelligent system called 'FireTrack', which combines Deep Neural Network (DNN) and Cellular Automata (CA) models.

This study delves into the dynamics of systems and their behaviors, particularly in the context of modeling forest fires. The document is organized into three chapters:

- **Chapter 1** lays the groundwork by introducing dynamic systems. It covers key concepts such as state variables (describing the system's current conditions), dynamics (how the system evolves over time), and equilibrium states (when the system is stable). These foundational ideas are crucial for understanding the advanced prediction techniques and research methods discussed in subsequent chapters.

- **Chapter 2** reviews prediction techniques for dynamic systems with a focus on forest fire prediction. It examines various machine learning models, highlighting their applications, strengths, and limitations. This review helps in selecting the most effective techniques for accurate fire predictions.

- **Chapter 3** is the core of the research, detailing the design and implementation of the proposed FireTrack framework. It explains the processes for data preparation, and the development of the CA model for simulating fire spread. Additionally, it describes the evaluation metrics used to assess the system's reliability and effectiveness. The results demonstrate significant improvements in prediction accuracy and fire spread simulation.

Future work will focus on enhancing the system by incorporating real-time data from sensors placed in forests. These sensors will continuously monitor environmental conditions such as temperature, humidity, and wind speed, allowing for more accurate and timely predictions. Future research will also explore advanced artificial intelligence techniques to develop strategies for effectively limiting the propagation of fires once detected.

Bibliography

[1] W Ross Ashby. Principles of the self-organizing dynamic system. *The Journal of general psychology*, 37(2):125–128, 1947.

[2] Adele M Hayes, Carly Yasinski, J Ben Barnes, and Claudi LH Bockting. Network destabilization and transition in depression: New methods for studying the dynamics of therapeutic change. *Clinical Psychology Review*, 41:27–39, 2015.

[3] Phil Hiver, Ali H Al-Hoorie, and Reid Evans. Complex dynamic systems theory in language learning: A scoping review of 25 years of research. *Studies in Second Language Acquisition*, 44(4):913–941, 2022.

[4] Diane Larsen-Freeman. On language learner agency: A complex dynamic systems theory perspective. *The modern language journal*, 103:61–79, 2019.

[5] Katsuhiko Ogata. *System dynamics*. 2004.

[6] Arnold J Sameroff. Development systems: Contexts and evolution. *Handbook of child psychology: formerly Carmichael's Manual of child psychology/Paul H. Mussen, editor*, 1983.

[7] Craig A Kluever. *Dynamic systems: modeling, simulation, and control*. John Wiley & Sons, 2020.

[8] Nilesh Satyavijay Pohokar, Sunil Ramchandra Gupta, Priyanka Anant Chorey, and Sonika Anant Chorey. *Fundamentals of research methodology (p. 77)*. RED'SHINE Publication, 2022.

[9] Soon Hoe Lim, Ludovico Theo Giorgini, Woosok Moon, and John S Wettlaufer. Predicting critical transitions in multiscale dynamical systems using reservoir computing. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 30(12), 2020.

[10] Harriet Black Nembhard and Ming Shu Kao. Adaptive forecast-based monitoring for dynamic systems. *Technometrics*, 45(3):208–219, 2003.

[11] Michael E Cholette, Jianbo Liu, Dragan Djurdjanovic, and Kenneth A Marko. Monitoring of complex systems of interacting dynamic systems. *Applied Intelligence*, 37:60–79, 2012.

[12] P Rajendra and V Brahmajirao. Modeling of dynamical systems through deep learning. *Biophysical Reviews*, 12(6):1311–1320, 2020.

[13] Maria Luminita Scutaru and Catalin-Iulian Pruncu. Mathematical modeling and simulation in mechanics and dynamic systems, 2024.

[14] Zeki Murat Çınar, Abubakar Abdussalam Nuhu, Qasim Zeeshan, Orhan Korhan, Mohammed Asmael, and Babak Safaei. Machine learning in predictive maintenance towards sustainable smart manufacturing in industry 4.0. *Sustainability*, 12(19):8211, 2020.

[15] Said Ćosić and István Vokony. Deep learning-based dynamic state estimation for frequency stability monitoring in power systems with high penetration of renewable generation. *Engineering Proceedings*, 41(1):16, 2023.

[16] Rui Wang and Rose Yu. Physics-guided deep learning for dynamical systems: A survey. *arXiv preprint arXiv:2107.01272*, 2021.

[17] Charles M Macal and Michael J North. Agent-based modeling and simulation. In *Proceedings of the 2009 winter simulation conference (WSC)*, pages 86–98. IEEE, 2009.

[18] Dirk Helbing. Agent-based modeling. In *Social self-organization: Agent-based simulations and experiments to study emergent social behavior*, pages 25–70. Springer, 2012.

[19] Jacques Ferber and Gerhard Weiss. *Multi-agent systems: an introduction to distributed artificial intelligence*, volume 1. Addison-wesley Reading, 1999.

[20] Steven F Railsback and Volker Grimm. *Agent-based and individual-based modeling: a practical introduction*. Princeton university press, 2019.

[21] Juan Martín García. *Agent-Based Modeling and Simulation I: Practical guide to the analysis of complex systems*. Juan Martín García, 2023.

[22] Changnian Han, Peng Zhang, Danny Bluestein, Guojing Cong, and Yuefan Deng. Artificial intelligence for accelerating time integrations in multiscale modeling. *Journal of computational physics*, 427:110053, 2021.

[23] Karl J Aström, Pedro Albertos, Mogens Blanke, Alberto Isidori, Walther Schaufelberger, and Ricardo Sanz. *Control of complex systems*. Springer Science & Business Media, 2011.

[24] Michael J Radzicki. System dynamics and its contribution to economics and economic modeling. *System dynamics: Theory and applications*, pages 401–415, 2020.

[25] M Hashem Pesaran. *Time series and panel data econometrics*. Oxford University Press, 2015.

[26] Jesús Crespo Cuaresma, Tapio Palokangas, and Alexander Tarasyev. *Dynamic systems, economic growth, and the environment*, volume 12. Springer Science & Business Media, 2009.

[27] Noelle E Selin, Amanda Giang, and William C Clark. Progress in modeling dynamic systems for sustainable development. *Proceedings of the National Academy of Sciences*, 120(40):e2216656120, 2023.

[28] Mathieu Andraud, Niel Hens, Christiaan Marais, and Philippe Beutels. Dynamic epidemiological models for dengue transmission: a systematic review of structural approaches. *PloS one*, 7(11):e49085, 2012.

[29] Rachel Colla, Paige Williams, Lindsay G Oades, and Jesus Camacho-Morles. "a new hope" for positive psychology: a dynamic systems reconceptualization of hope theory. *Frontiers in Psychology*, 13:809053, 2022.

[30] Jing Li, Valerij Romanovskij, Xiaodong Yang, Yong Xu, Kuan Lu, and Shaotao Zhu. Introduction: Theory and applications of dynamical systems in engineering interdiscipline. *Journal of Nonlinear Mathematical Physics*, 29(2):187–189, 2022.

[31] Ying He, Xinyuan Liang, Menglu Chen, Ting Tian, Yimeng Zeng, Jin Liu, Lei Hao, Jiahua Xu, Rui Chen, Yanpei Wang, et al. Development of brain state dynamics involved in working memory. *Cerebral Cortex*, 33(11):7076–7087, 2023.

[32] Kashif Abbass, Muhammad Zeeshan Qasim, Huaming Song, Muntasir Murshed, Haider Mahmood, and Ijaz Younis. A review of the global climate change impacts, adaptation, and sustainable mitigation measures. *Environmental Science and Pollution Research*, 29(28):42539–42559, 2022.

[33] Christian Janiesch, Patrick Zschech, and Kai Heinrich. Machine learning and deep learning. *Electronic Markets*, 31(3):685–695, 2021.

[34] Feng-Jen Yang. An extended idea about decision trees. In *2019 international conference on computational science and computational intelligence (CSCI)*, pages 349–354. IEEE, 2019.

[35] Shiyang Xuan, Guanjun Liu, Zhenchuan Li, Lutao Zheng, Shuo Wang, and Changjun Jiang. Random forest for credit card fraud detection. In *2018 IEEE 15th international conference on networking, sensing and control (ICNSC)*, pages 1–6. IEEE, 2018.

[36] Lucia Alessi and Roberto Savona. Machine learning for financial stability. In *Data Science for Economics and Finance: Methodologies and Applications*, pages 65–87. Springer International Publishing Cham, 2021.

[37] Priyanka and Dharmender Kumar. Decision tree classifier: a detailed survey. *International Journal of Information and Decision Sciences*, 12(3):246–269, 2020.

[38] Rituparna Mukherjee and Abhinandan De. Real-time dynamic security analysis of power systems using strategic pmu measurements and decision tree classification. *Electrical Engineering*, 103(2):813–824, 2021.

[39] Surat Asvapoositkul and Robin Preece. Decision tree-based prediction model for small signal stability and generation-rescheduling preventive control. *Electric Power Systems Research*, 196:107200, 2021.

[40] Srabanti Maji and Srishti Arora. Decision tree algorithms for prediction of heart disease. In *Information and Communication Technology for Competitive Strategies: Proceedings of Third International Conference on ICTCS 2017*, pages 447–454. Springer, 2019.

[41] Kanak Saxena, Richa Sharma, et al. Efficient heart disease prediction system using decision tree. In *International Conference on Computing, Communication & Automation*, pages 72–77. IEEE, 2015.

[42] Trevor Hastie, Robert Tibshirani, Jerome H Friedman, and Jerome H Friedman. *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer, 2009.

[43] Markus HA Janse, Fons van der Sommen, Svitlana Zinger, Erik J Schoon, et al. Early esophageal cancer detection using rf classifiers. In *Medical imaging 2016: computer-aided diagnosis*, volume 9785, pages 344–351. SPIE, 2016.

[44] Zuherman Rustam and Glori Stephani Saragih. Predicting bank financial failures using random forest. In *2018 International Workshop on Big Data and Information Security (IWBIS)*, pages 81–86. IEEE, 2018.

[45] Lin Zhu, Dafeng Qiu, Daji Ergu, Cai Ying, and Kuiyi Liu. A study on predicting loan default based on the random forest algorithm. *Procedia Computer Science*, 162:503–513, 2019.

[46] Celestine Iwendi, Ali Kashif Bashir, Atharva Peshkar, R Sujatha, Jyotir Moy Chatterjee, Swetha Pasupuleti, Rishita Mishra, Sofia Pillai, and Ohyun Jo. Covid-19 patient health prediction using boosted random forest algorithm. *Frontiers in public health*, 8:357, 2020.

[47] Dmytro Chumachenko, Ievgen Meniailov, Kseniia Bazilevych, Tetyana Chumachenko, and Sergey Yakovlev. Investigation of statistical machine learning models for covid-19 epidemic process simulation: Random forest, k-nearest neighbors, gradient boosting. *Computation*, 10(6):86, 2022.

[48] Alexandra Chouldechova, Diana Benavides-Prado, Oleksandr Fialko, and Rhema Vaithianathan. A case study of algorithm-assisted decision making in child maltreatment hotline screening decisions. In *Conference on Fairness, Accountability and Transparency*, pages 134–148. PMLR, 2018.

[49] Imen Charfi, Johel Miteran, Julien Dubois, Mohamed Atri, and Rached Tourki. Definition and performance evaluation of a robust svm based fall detection solution. In *2012 eighth international conference on signal image technology and internet based systems*, pages 218–224. IEEE, 2012.

[50] K Srunitha and S Padmavathi. Performance of svm classifier for image based soil classification. In *2016 International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES)*, pages 411–415. IEEE, 2016.

[51] Narendra S Chaudhari, Arana Tiwari, and Jaya Thomas. A novel svm based approach for noisy data elemination. In *2010 11th International Conference on Control Automation Robotics & Vision*, pages 1760–1765. IEEE, 2010.

[52] Xiao Liang, Taiyue Qi, Zhiyi Jin, and Wangping Qian. Hybrid support vector machine optimization model for inversion of tunnel transient electromagnetic method. *Math. Biosci. Eng*, 17(4):3998–4017, 2020.

[53] Vijander Singh, Ramesh Chandra Poonia, Sandeep Kumar, Pranav Dass, Pankaj Agarwal, Vaibhav Bhatnagar, and Linesh Raja. Prediction of covid-19 corona virus pandemic based on time series data using support vector machine. *Journal of Discrete Mathematical Sciences and Cryptography*, 23(8):1583–1597, 2020.

[54] Sofiane Atek, Filippo Bianchini, Corrado De Vito, Vincenzo Cardinale, Simone Novelli, Cristiano Pesaresi, Marco Eugeni, Massimo Mecella, Antonello Rescio, Luca Petronzio, et al. A predictive decision support system for coronavirus disease 2019 response management and medical logistic planning. *Digital Health*, 9:20552076231185475, 2023.

[55] Iqbal H Sarker. Deep learning: a comprehensive overview on techniques, taxonomy, applications and research directions. *SN Computer Science*, 2(6):420, 2021.

[56] Faroudja Abid. A survey of machine learning algorithms based forest fires prediction and detection systems. *Fire technology*, 57(2):559–590, 2021.

[57] Sushrut Gaikwad, Bipin Kumar, Prafull P Yadav, Rupal Ambulkar, Gaurav Govardhan, Santosh H Kulkarni, Rajesh Kumar, Dilip M Chate, Narendra Nigam, Suryachandra A Rao, et al. Harnessing deep learning for forecasting fire-burning locations and unveiling pm 2.5 emissions. *Modeling Earth Systems and Environment*, 10(1):927–941, 2024.

[58] Weibo Liu, Zidong Wang, Xiaohui Liu, Nianyin Zeng, Yurong Liu, and Fuad E Alsaadi. A survey of deep neural network architectures and their applications. *Neurocomputing*, 234:11–26, 2017.

[59] Guangyuan Liao and Limin Zhang. Solving flows of dynamical systems by deep neural networks and a novel deep learning algorithm. *Mathematics and Computers in Simulation*, 202:331–342, 2022.

[60] Tong Qin, Kailiang Wu, and Dongbin Xiu. Data driven governing equations approximation using deep neural networks. *Journal of Computational Physics*, 395:620–635, 2019.

[61] Peng Wang and Fengqi Si. Dynamic prediction of the thermal nonlinear process based on deep hybrid neural network. In *E3S Web of Conferences*, volume 162, page 01007. EDP Sciences, 2020.

[62] Chiou-Jye Huang, Yamin Shen, Yung-Hsiang Chen, and Hsin-Chuan Chen. A novel hybrid deep neural network model for short-term electricity price forecasting. *International Journal of Energy Research*, 45(2):2511–2532, 2021.

[63] Jie Shi, Brandon Foggo, and Nanpeng Yu. Power system event identification based on deep neural network with information loading. *IEEE Transactions on Power Systems*, 36(6):5622–5632, 2021.

[64] Jingyu Zhao, Feiqing Huang, Jia Lv, Yanjie Duan, Zhen Qin, Guodong Li, and Guangjian Tian. Do rnn and lstm have long memory? In *International Conference on Machine Learning*, pages 11365–11375. PMLR, 2020.

[65] Dong-Hoon Shin, Kyungyong Chung, and Roy C Park. Prediction of traffic congestion based on lstm through correction of missing temporal and spatial data. *IEEE Access*, 8:150784–150796, 2020.

[66] Jie Sun and Jiwon Kim. Joint prediction of next location and travel time from urban vehicle trajectories using long short-term memory neural networks. *Transportation Research Part C: Emerging Technologies*, 128:103114, 2021.

[67] Libo Zhang, Huaxiong Li, Xianzhong Zhou, and Bing Huang. Sequential three-way decision based on multi-granular autoencoder features. *Information sciences*, 507:630–643, 2020.

[68] Chong Zhou and Randy C Paffenroth. Anomaly detection with robust deep autoencoders. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 665–674, 2017.

[69] Andri Ashfahani, Mahardhika Pratama, Edwin Lughofer, and Yew-Soon Ong. Devdan: Deep evolving denoising autoencoder. *Neurocomputing*, 390:297–314, 2020.

[70] Sangmin Lee, Hae Joong Kim, and Seoung Bum Kim. Dynamic dispatching system using a deep denoising autoencoder for semiconductor manufacturing. *Applied Soft Computing*, 86:105904, 2020.

[71] Sean Givnan, Carl Chalmers, Paul Fergus, Sandra Ortega-Martorell, and Tom Whalley. Anomaly detection using autoencoder reconstruction upon industrial motors. *Sensors*, 22(9):3166, 2022.

[72] Mashrur Chowdhury and Adel W Sadek. Advantages and limitations of artificial intelligence. *Artificial intelligence applications to critical transportation issues*, 6(3):360–375, 2012.

[73] Kendall Preston Jr and Michael JB Duff. *Modern cellular automata: theory and applications*. Springer Science & Business Media, 2013.

[74] Piotr Podolski and Hung Son Nguyen. Cellular automata in covid-19 prediction. *Procedia Computer Science*, 192:3370–3379, 2021.

[75] Mozhdeh Khaksar Manshad, Mohammad Reza Meybodi, and Afshin Salajegheh. A new irregular cellular learning automata-based evolutionary computation for time series link prediction in social networks. *Applied Intelligence*, 51(1):71–84, 2021.

[76] Inés Santé, Andrés M García, David Miranda, and Rafael 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.

[77] Guy Engelen, Roger White, and Inge Uljee. Integrating constrained cellular automata models, gis and decision support tools for urban planning and policy-making. In *Decision support systems in urban planning*, pages 113–135. Routledge, 2003.

[78] Yangzexi Liu, Jingqiu Guo, John Taplin, and Yibing Wang. Characteristic analysis of mixed traffic flow of regular and autonomous vehicles using cellular automata. *Journal of Advanced Transportation*, 2017(1):8142074, 2017.

[79] Lemont B Kier, Chao-Kun Cheng, and Bernard Testa. Cellular automata models of biochemical phenomena. *Future Generation Computer Systems*, 16(2-3):273–289, 1999.

[80] Pedro HT 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*, 200:105832, 2021.

[81] Ross D Collins, Richard de Neufville, João Claro, Tiago Oliveira, and Abílio P Pacheco. Forest fire management to avoid unintended consequences: A case study of portugal using system dynamics. *Journal of environmental management*, 130:1–9, 2013.

[82] Farshad Farkhondehmaal. *Wildfire as Coupled Human Natural System*. PhD thesis, Virginia Tech, 2022.

[83] Christian Messier and Klaus J Puettmann. Forests as complex adaptive systems: implications for forest management and modelling. *L'Italia forestale e montana*, 66(3):249–258, 2011.

[84] Daniela Stojanova, Panče Panov, Andrej Kobler, Sašo Džeroski, and Katerina Taškova. Learning to predict forest fires with different data mining techniques. In *Conference on data mining and data warehouses (SiKDD 2006), Ljubljana, Slovenia*, pages 255–258. Citeseer, 2006.

[85] Slobodan Milanović, Nenad Marković, Dragan Pamučar, Ljubomir Gigović, Pavle Kostić, and Sladjan D Milanović. Forest fire probability mapping in eastern serbia: Logistic regression versus random forest method. *Forests*, 12(1):5, 2020.

[86] Xin Wu, Gui Zhang, Zhigao Yang, Sanqing Tan, Yongke Yang, and Ziheng Pang. Machine learning for predicting forest fire occurrence in changsha: An innovative investigation into the introduction of a forest fuel factor. *Remote Sensing*, 15(17):4208, 2023.

[87] Gopi Battineni, Nalini Chintalapudi, and Francesco Amenta. Machine learning in medicine: Performance calculation of dementia prediction by support vector machines (svm). *Informatics in Medicine Unlocked*, 16:100200, 2019.

[88] Dao Wen Xie and Shi Liang Shi. Prediction for burned area of forest fires based on svm model. *Applied Mechanics and Materials*, 513:4084–4089, 2014.

[89] Gözde Bayat and Kazım Yıldız. Comparison of the machine learning methods to predict wildfire areas. *Turkish Journal of Science and Technology*, 17(2):241–250, 2022.

[90] Paulo Cortez and Aníbal de Jesus Raimundo Morais. A data mining approach to predict forest fires using meteorological data. 2007.

[91] Cheng-Yu Ku and Chih-Yu Liu. Predictive modeling of fire incidence using deep neural networks. *Fire*, 7(4):136, 2024.

[92] Can Lai, Shucai Zeng, Wei Guo, Xiaodong Liu, Yongquan Li, and Boyong Liao. Forest fire prediction with imbalanced data using a deep neural network method. *Forests*, 13(7):1129, 2022.

[93] Sam-Keun Kim and Jae-Geun Ahn. Data mining based forest fires prediction models using meteorological data. *Journal of the Korea Academia-Industrial cooperation Society*, 21(8):521–529, 2020.

[94] Hao Liang, Meng Zhang, and Hailan Wang. A neural network model for wildfire scale prediction using meteorological factors. *IEEE Access*, 7:176746–176755, 2019.

[95] Saily Natekar, Shivani Patil, Aishwarya Nair, and Sukanya Roychowdhury. Forest fire prediction using lstm. In *2021 2nd International Conference for Emerging Technology (INCET)*, pages 1–5. IEEE, 2021.

[96] Narayanan Ganesh, Rajendran Shankar, Miroslav Mahdal, Janakiraman Senthil Murugan, Jasgurpreet Singh Chohan, and Kanak Kalita. Exploring deep learning methods for computer vision applications across multiple sectors: Challenges and future trends. *CMES-Computer Modeling in Engineering & Sciences*, 139(1), 2024.

[97] Zulfiqar Ahmad Khan, Tanveer Hussain, Fath U Min Ullah, Suneet Kumar Gupta, Mi Young Lee, and Sung Wook Baik. Randomly initialized cnn with densely connected stacked autoencoder for efficient fire detection. *Engineering Applications of Artificial Intelligence*, 116:105403, 2022.

[98] Sara Aleixo Perestrelo, Maria Clara Grácio, Nuno de Almeida Ribeiro, and Luís Mário Lopes. A multi-scale network with percolation model to describe the spreading of forest fires. *Mathematics*, 10(4):588, 2022.

[99] Shan HUI, Xiaoping RUI, and Yao LI. An improved forest fire spread simulation algorithm coupled with cellular automata. *Geomatics and Information Science of Wuhan University*, 41(10):1326–1332, 2016.

[100] Navid Mahdizadeh Gharakhanlou and Navid Hooshangi. Dynamic simulation of fire propagation in forests and rangelands using a gis-based cellular automata model. *International journal of wildland fire*, 30(9):652–663, 2021.

[101] O Jellouli, A Bernoussi, M Mâatouk, and M Amharref. Forest fire modelling using cellular automata: application to the watershed oued laou (morocco). *Mathematical and computer modelling of dynamical systems*, 22(5):493–507, 2016.

[102] Xuan Sun, Ning Li, Duoqi Chen, Guang Chen, Changjun Sun, Mulin Shi, Xuehong Gao, Kuo Wang, and Ibrahim M Hezam. A forest fire prediction model based on cellular automata and machine learning. *IEEE Access*, 2024.

[103] Ying Zhi, Yao Jiang, Diwen Ke, Xianlei Hu, and Xianghua Liu. Review on cellular automata for microstructure simulation of metallic materials. *Materials*, 17(6):1370, 2024.

[104] Piyush Jain, Sean CP Coogan, Sriram Ganapathi Subramanian, Mark Crowley, Steve Taylor, and Mike D Flannigan. A review of machine learning applications in wildfire science and management. *Environmental Reviews*, 28(4):478–505, 2020.

[105] Binh Thai Pham, Abolfazl Jaafari, Mohammadtaghi Avand, Nadhir Al-Ansari, Tran Dinh Du, Hoang Phan Hai Yen, Tran Van Phong, Duy Huu Nguyen, Hiep Van Le, Davood Mafi-Gholami, et al. Performance evaluation of machine learning methods for forest fire modeling and prediction. *Symmetry*, 12(6):1022, 2020.

[106] M Arif, KK Alghamdi, SA Sahel, SO Alosaimi, ME Alsahaft, MA Alharthi, and M Arif. Role of machine learning algorithms in forest fire management: A literature review. *J. Robot. Autom*, 5:212–226, 2021.

[107] Ramez Alkhatib, Wahib Sahwan, Anas Alkhatieb, and Brigitta Schütt. A brief review of machine learning algorithms in forest fires science. *Applied Sciences*, 13(14):8275, 2023.

[108] Loubna Seddiki. Intelligent models for dynamic systems monitoring: A case study on forest fires. In *AI 24 DAY, Guelma University*, 2024.

[109] Michal S Gal and Daniel L Rubinfeld. Data standardization. *NYUL Rev.*, 94:737, 2019.

[110] Yun Xu and Royston Goodacre. On splitting training and validation set: a comparative study of cross-validation, bootstrap and systematic sampling for estimating the generalization performance of supervised learning. *Journal of analysis and testing*, 2(3):249–262, 2018.

[111] Sebastian Raschka. Model evaluation, model selection, and algorithm selection in machine learning. *arXiv preprint arXiv:1811.12808*, 2018.

[112] Liyuan Liu, Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Jiawei Han. On the variance of the adaptive learning rate and beyond. *arXiv preprint arXiv:1908.03265*, 2019.

[113] Wes McKinney et al. Data structures for statistical computing in python. In *SciPy*, volume 445, pages 51–56, 2010.

[114] Uri Wilensky. Netlogo. evanston, il: Center for connected learning and computer-based modeling, northwestern university, 1999.

[115] Dalia Shihab Ahmed, Rasha shaker ibrahim Al-badri, Firas Ali Hashim, and Nadia Mahmood Hussien. Keras deep learning package in python: A review. *European Journal of Interdisciplinary Research and Development*, 19:24–29, 2023.

[116] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.

[117] Stefan Van Der Walt, S Chris Colbert, and Gael Varoquaux. The numpy array: a structure for efficient numerical computation. *Computing in science & engineering*, 13(2):22–30, 2011.

[118] John D Hunter. Matplotlib: A 2d graphics environment. *Computing in science & engineering*, 9(03):90–95, 2007.

Webographie

[W1] , https://eminebozkus.medium.com/machine-learning-101-a-beginners-guide-to-understanding-the-basics-d8a85ff2c8e, Last access to the website : 23/04/2024.

[W2] , https://www.javatpoint.com/machine-learning-decision-tree-classification-algorithm, Last access to the website : 29/04/2024.

[W3] , https://www.geeksforgeeks.org/random-forest-algorithm-in-machine-learning/, Last access to the website : 08/05/2024.

[W4] , https://vitalflux.com/classification-model-svm-classifier-python-example/, Last access to the website : 08/05/2024.

[W5] , https://iotric.medium.com/deep-neural-network-dnn-its-scope-and-nature-of-complexity-56af59f87ea4, Last access to the website : 12/05/2024.

[W6] , https://commons.wikimedia.org/wiki/File:The_LSTM_Cell.svg, Last access to the website : 20/05/2024.

[W7] , https://www.geeksforgeeks.org/types-of-autoencoders/, Last access to the website : 29/05/2024.

[W8] , https://www.javatpoint.com/php-tutorial, Last access to the website : 05/06/2024.

[W9] , https://code.visualstudio.com/docs/editor/whyvscode, Last access to the website : 05/06/2024.

[W10] , https://www.infoworld.com/article/3666488/what-is-visual-studio-code-microsofts-extensible-code-editor.html, Last access to the website : 05/06/2024.

[W11] , https://www.javatpoint.com/xampp, Last access to the website : 05/06/2024.

[W12] , https://www.talend.com/resources/what-is-mysql/, Last access to the website : 05/06/2024.

[W13] , Abid, Faroudja. (2019). Algerian Forest Fires. UCI Machine Learning Repository. https://archive.ics.uci.edu/dataset/547/algerian+forest+fires+dataset/, Last access to the website : 15/06/2024.

[W14] , https://pandas.pydata.org/docs/, Last access to the website : 16/06/2024.

[W15] , https://wiki.python.org/moin/PyQt/ Last access to the website : 16/06/2024.