



People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research

IBN KHALDOUN UNIVERSITY OF TIARET

Dissertation

Presented to:

FACULTY OF MATHEMATICS AND COMPUTER SCIENCE
DEPARTEMENT OF MATHEMATICS

in order to obtain the degree of :

MASTER

Specialty: **Computer Engineering**

Presented by:

BOUABIB OUMELKHEIR
BENMENAOUARA SARA

On the theme:

Modeling the Spread of an Epidemic Using Artificial Intelligence

Defended publicly on ... /... /2024 in Tiaret in front the jury composed of:

Mr BENGHENI Abdelmalek	MCA	Tiaret University	Chairman
Mr MEBAEK Bendaoud	Pr	Tiaret University	Supervisor
Mr M'assakri Mustapha	MCA	Tiaret University	Examiner

2024-2025

Thanks

"First and foremost, all glory and thanks to God for His unwavering grace and guidance throughout this journey. Without His strength, none of this would have been possible.

Thanks to our amazing parents, thank you for your endless love, sacrifices, and encouragement. You've been our biggest supporters, and we are forever grateful for everything you've done to help us reach this milestone.

Thanks To all the brothers, and sisters, thank you for always being there, cheering on, and keeping grounded. Your love and laughter have been our motivation. This achievement is as much yours as it is ours.

Thanks to our supervisor Mr. Mebarek Bendaoud who helped us a lot and supported us in our research

With a heart full of gratitude, thank you all for being foundation and inspiration!"

Dedicate

To the beacon of my life, to an inexhaustible tenderness, to my precious mother...

I dedicate this humble effort to my beloved mother, who was my support and help, and who sacrificed everything precious and precious for me. To the one who taught me the meaning of patience and determination, to the one whose warm words motivated me in the most difficult moments. Mum. All my love and gratitude to you, you are the roots of the success I am gaining today.

I do not forget my dear father, who was the best helper and support. Thank you for all the sacrifices, for all the support and dedication. I dedicate to you the fruit of my labor, for you are the reason for my existence and the reason for my success.

My beloved brothers Tuati and Benahmed, my sisters Tamtam, Babika, Sherine and my Eline. This achievement is the beginning of the path, and it is the result of your endless love and prayers. May your hands be blessed, may your giving last, and I ask God to live up to your expectations.

With all love and gratitude

Oumelkhier

Dedicate

I would like to express my deepest and most heartfelt gratitude to my mother.

Mama, your love, strength, and constant support have been the foundation of everything I've achieved. You have sacrificed so much for me, and your faith in me

has given me the courage to keep going, even in the most difficult moments.

This work is as much yours as it is mine. I am forever grateful to be your child.

Thank you, Mama,

I also want to thank my father from the depths of my heart. Papa, your strength, wisdom, and quiet support have always been a guiding light in my life. You've taught me the value of hard work, resilience, and integrity lessons that have shaped who I

am today.

Even in silence, your presence gave me courage. Thank you for always standing by my side.

I am also sincerely thankful to my brothers and sisters for their love, understanding, and for always being there for me. Their presence in my life has been a source of strength and joy

SARA

Abstract

The COVID-19 pandemic has demonstrated the critical importance of accurate and timely epidemic modeling to guide public health responses. Traditional compartmental models such as SIR and SEIR, while effective in capturing fundamental transmission dynamics, often rely on fixed parameters and assumptions that may not hold in complex, real-world scenarios. In contrast, artificial intelligence (AI) offers a data-driven alternative capable of learning from vast and evolving datasets. This study explores the application of AI techniques specifically deep learning models such as Long Short-Term Memory (LSTM) and Bidirectional LSTM (BiLSTM) to forecast the spread of COVID-19. These models are trained on time-series data including daily case counts, mobility trends, and government intervention measures.

This study explores the application of artificial intelligence (AI) techniques including machine learning (ML) and deep learning (DL) to model the spread of COVID-19. By leveraging AI-driven approaches such as recurrent neural networks (RNNs), long short-term memory (LSTM) models, and Bidirectional LSTM Model.

The objective of artificial intelligence (AI) research is to develop computer systems capable of performing tasks using thinking processes similar to those of human beings. Thus, the challenge lies not only in understanding human thought, but also in its modelling and reproduction.

The use of CNNs and RNNs was an optimal solution given the results obtained in several research projects. Results demonstrate that AI models can capture nonlinear patterns and temporal dependencies more effectively than traditional models, enabling improved short-term forecasting accuracy. Furthermore, we propose a hybrid modeling framework that integrates mechanistic and deep learning methods to leverage the interpretability of epidemiological models and the predictive power of AI. The findings underscore the potential of AI to enhance epidemic preparedness and real-time decision-making during public health crises.

Keywords: COVID-19, SIR-SEIR, epidemic modeling, artificial intelligence, machine learning, deep learning, LSTM-BiLSTM, predictive analytics, public health

Table of contents

Thanks

Dedicate

Abstract

General introduction 1

Chapter I: Mathematical modeling of the spread of epidemics

Part I: Communicable diseases and epidemiology 6

1. Background and importance of modeling the spread of epidemics 6

1.1. Background of modeling the spread of epidemics 8

1.2. The importance of modeling the spread of epidemics 8

2. The impact of epidemics on public health and crisis management systems 8

2.1. The impact on public health 8

2.2. The impact on crisis management system 9

3. Challenges of classical models in the face of dynamic epidemics 9

Part II: Communicable diseases and epidemiology 9

Mathematical modeling 9

1. Presentation of Classical Epidemiological Models SIR, SEIR, and their variants 9

2. Differential equations used, assumptions, and conditions of application 12

2.1. SIR Model (Susceptible, Infected, Recovered) 12

2.2. SEIR Model (Susceptible, Exposed, Infected, Recovered) 13

2.3. Key Differences between SIR and SEIR Models 15

3. Practical Applications of the Models and Analysis of Their Limitations 15

3.1. Practical application to the SIR model 15

3.2. Practical application to the SEIR model 16

3.3. The limitations of the models 17

Hybrid models and the role of AI in modeling 17

1. Introduction to Hybrid models 17

2. An overview of artificial intelligence in epidemiology	18
3. Comparison of classical approaches and AI-based approaches	20
3.1. Accuracy and Efficiency: Comparing Detection Rates and False Positives / Negatives	20
3.2. Computational and Data Needs for Traditional vs. AI Approaches	21
Conclusion	22

Chapter II: IA approach, Deep-learning, LSTM and BiLSTM

I. IA approach and Deep-learning	24
1. Definition of machine learning (Machine Learning)	25
2. Type of learning	25
3. Deep Learning Definition (DL)	26
4. Neural Networks	27
4.1. Biological neuron	27
4.2. Principle of neural networks (Neural Network)	28
4.3. Functioning of Neural Networks	28
5. Types of Neural Network	29
5.1. Artificial Neural Networks (ANN)	29
5.1.1. Artificial neural network architecture	29
5.1.2. Perceptron	30
5.2. Recurrent Neural Networks (RNN)	31
5.2.1. Hopfield Networks	32
5.2.2. LSTM	33
5.2.3. GRU	33
5.2.4. JANET	33
5.3. Convolution Neural Networks (CNN)	34
5.3.1. CNN Diapers	34
5.3.2. The backpropagation algorithm	35
II. LSTM and BiLSTM	35

1. LSTM	35
1.1. Metrics	36
1.2. Definition of Long Short-Term Memory (LSTM)	38
1.3. The architecture of an LSTM model	39
1.3.1. LSTM Weights	39
1.3.2. LSTM Doors (Gated)	39
2. Bidirectional LSTM Model	42
2.1. What is Bi-LSTM and how it works	42
2.2. BiLSTM architecture	44
Conclusion	46

Chapter III: Modeling Approach

1. Introduction	48
2. Dataset	48
3. SIR model	49
3.1. Initial Conditions	49
3.1.1. Loss Function and Optimization	50
3.1.2. Weighted Mean Squared Error (MSE) Loss Function	50
3.1.3. Optimization of Model Parameters	52
3.1.4. Differential Evolution Algorithm	52
3.2. Simulation and Prediction	53
3.2.1. Comparison with Actual Data	55
3.3. Results and Analysis	55
3.3.1. Visualization of Predictions	56
3.3.2. Model Evaluation: R^2 Score	56
3.3.3. Key Extracted Metrics	57
4. Deep learning model LSTM	58
4.1. Dataset and Feature Engineering	58
4.2. Data Preprocessing	58

4.2.1. Feature Scaling	58
4.2.2. Sequence Creation	59
4.3. Model Architecture	60
4.3.1. Bidirectional LSTM Network	60
4.3.2. Layer Configuration.	60
4.4. Training Strategy	62
4.4.1. Optimization Algorithm	62
4.4.2. Loss Function	63
4.4.3. Training Process Enhancement	63
4.4.4. Training Configuration	64
4.5. Model Evaluation	65
4.5.1. Performance Metrics	65
4.5.2. Visualization	65
5. Comparison with SIR Model	66
6. Conclusion	68

Chapter IV: Application Architecture and Implementation

1. Introduction	70
2. Application Overview	70
2.1. Sidebar Controls	70
2.2. Data overview	71
2.3. Get prediction	72
2.4. Comparison between the results of the models	74
2.5. Interactive Report Generation	75
3. Technologies Used	78
3.1. Core Framework	78
3.2. Data Processing and Analysis	78
3.3. Deep learning	80
3.4. Visualization	81

4. Future Enhancements	82
5. Conclusion	83
General conclusion	84
References	86

List of figures

Figure 1: Growth of infection and depletion of the susceptible population in an SIR outbreak	11
Figure 2: transmission diagram for the SIR model	12
Figure 3 transmission diagram for the SEIR model	14
Figure 4: Applications of artificial intelligence in public health systems	18
Figure 5: Diagram of the different use cases for a given type of training	26
Figure 6: Structure of a biological neuron	27
Figure 7: The architecture of an artificial neural network	30
Figure 8 : La structure d'un neurone artificiel	31
Figure 9: CNN's diagram	35
Figure 10: LSTM doors	40
Figure 11: LSTM architecture	41
Figure 12: Basic structure of a BiLSTM model	42
Figure 13: BiLSTM model showing the input and output layers	44
Figure 14: BiLSTM architecture	45
Figure 15: SIR model	54
Figure 16: LSTM model architecture	61
Figure 17: Data overview page	71
Figure 18: The SIR model prediction	72

Figure 19: the deep learning model prediction	73
Figure 20: Graphical comparison Active cases &deaths	74
Figure 21: Graphical of confirmed cases comparison	75
Figure 22: Graphical of Recovered cases comparis	75
Figure 23: Textual comparison	76
Figure 24: The final report	77
Figure 25: Streamlit	78
Figure 26: Pandas	78
Figure 27: NumPy	79
Figure 28: SciPy	79
Figure 29: TensorFlow	80
Figure 30: Scikit-learn	80
Figure 31: Matplotlib	81
Figure 32: Seaborn	81

List of tables

Table 1: comparison of malware detection methods: accuracy and false positives/ negatives	20
Table 2: computational and data requirements of malware detection methods	21
Table 3: Comparison between SIR and Deep Learning approaches	67

General introduction

General introduction

The rapid spread of the COVID-19 pandemic around the world has highlighted the crucial importance of forecasting and decision support tools for health authorities. Understanding and anticipating the evolution of an epidemic is a major challenge to limit its health, economic and social impacts. Traditionally, this task has been based on classical epidemiological models such as SIR or SEIR, which divide the population into compartments according to health status. However, these approaches may be limited in the face of the complexity of the real dynamics of an epidemic, especially when it comes to integrating multiple factors such as population mobility, variability of behavior, or the heterogeneity of geographical contexts.

In this context, artificial intelligence (AI), particularly machine learning techniques, offers new perspectives for the modelling of infectious disease spread. Thanks to their ability to analyze large amounts of heterogeneous data (epidemiological data, mobility data, climate data, etc.) and extract complex patterns from them, these tools allow the construction of powerful and adaptive predictive models. Applied to the COVID-19 pandemic, AI has been used in many projects to predict changes in the number of cases, identify areas at risk, or evaluate the effectiveness of containment measures.

This work is part of this dynamic by exploring the contributions of artificial intelligence to modeling the spread of COVID-19. It will analyze existing approaches, understand their methodological underpinnings, and assess their relevance to the challenges posed by a global health crisis.

We divided this research topic into 4 chapters:

We discussed it in the **first chapter** Mathematical modeling of the spread of epidemics involves using mathematical frameworks and equations to describe how infectious diseases propagate through populations over time. These models help predict the course of an epidemic, evaluate intervention strategies, and understand the dynamics of disease transmission.

We discussed it in the **chapter 2** moving into AI approaches for modeling epidemics, specifically deep learning, and recurrent neural networks like LSTM and BiLSTM. These techniques are data-driven and powerful for modeling time-series data, such as the daily number of infections, recoveries, or deaths.

In the **third chapter** we talk about how to build a robust model of an epidemic (COVID-19), we can take several modeling approaches, depending on our goal (e.g., understanding transmission vs. forecasting cases) and the data available

Finally in **the last chapter** we discussed it how to create a real-world application for epidemic modeling (e.g., COVID-19 tracker + predictor), we need a layered architecture that includes data ingestion, modeling, prediction, visualization, and deployment

Research questions

As part of the AI modelling of the spread of COVID-19, several research questions emerge to guide the analysis and evaluation of AI approaches. These questions make it possible to explore the challenges, performance and implications of predictive models. The following is a structured list of key issues:

- What are the most effective AI algorithms for predicting the spread of COVID-19? (Comparison between neural networks, agent models, random forests...)
- How do hybrid models (combining AI and conventional epidemiology) improve the accuracy of predictions?
- What is the impact of different data sources (mobility, social networks, and clinical data) on AI model performance?
- What's the role of deep learning, LSTM and BiLSTM for in epidemic predictions?
- What are the risks associated with the use of AI in public health (algorithmic bias, data protection, mass surveillance)?
- These questions help structure research around the technical, practical and ethical aspects of COVID-19 AI modelling, while opening up avenues for future improvements and applications in public health.

Purpose

The study of artificial intelligence (AI)-driven modeling for COVID-19 propagation serves several critical purposes in public health, data science, and epidemic preparedness:

- Enhancing Predictive Accuracy for Outbreak Management

Traditional epidemiological models (e.g., SIR, SEIR) rely on fixed parameters, whereas AI can analyze vast, dynamic datasets (mobility patterns, social media, climate, and genomic data) to improve real-time forecasting.

Machine learning (ML) models can detect nonlinear transmission patterns, helping predict infection waves, hospitalizations, and deaths more accurately.

- **Optimizing Public Health Interventions**

AI can simulate the impact of different policies (lockdowns, mask mandates, vaccination campaigns) to guide decision-making.

Reinforcement learning and agent-based modeling help identify optimal containment strategies while minimizing economic and social disruptions.

- **Early Detection and Surveillance**

AI-powered tools (e.g., natural language processing for social media, computer vision for mask compliance) enable early outbreak detection before traditional reporting.

Predictive models can flag emerging variants by analyzing viral genomic data and global spread trends.

- **Resource Allocation and Healthcare Planning**

AI helps forecast ICU bed demand, ventilator needs, and vaccine distribution to prevent healthcare system overload.

Reinforcement learning can optimize testing strategies and contact tracing efficiency.

Description

Modeling the Spread of an Epidemic Using Artificial Intelligence: COVID19 explores the application of AI techniques such as machine learning, deep learning, and agent-based simulations to predict and analyze the transmission dynamics of COVID-19. The subject covers data-driven modeling, risk assessment, intervention strategies, and real-time outbreak forecasting, highlighting how AI enhances epidemic response and public health decision-making. Topics include neural networks, epidemiological models, and the challenges of integrating AI with traditional disease control methods.

Forecast:

The integration of Artificial Intelligence (AI) in modeling the spread of COVID-19 will continue to advance, improving epidemic prediction accuracy and real-time response. Future developments may include:

- Enhanced predictive models using deep learning and hybrid AI-epidemiological approaches for early outbreak detection.
- AI-driven personalized risk assessment to optimize public health interventions.
- Integration with big data (e.g., mobility patterns, genomic surveillance) for dynamic transmission tracking.
- Challenges such as data biases, model interpretability, and ethical concerns will require ongoing research.

AI is expected to play a critical role in future pandemic preparedness, enabling faster, data-informed decision-making for global health security.

Chapter I

Mathematical modeling of the spread of epidemics

Introduction

Mathematical models, visualize how infectious diseases progress to show the likely outcome of an epidemic and help guide public health interventions. Models use underlying assumptions or aggregate statistics, along with mathematics, to obtain parameters for different infectious diseases. Using these parameters, they calculate the effects of different interventions, such as mass vaccination programs. Modeling may help determine which interventions to avoid and which to try, or it can predict future growth patterns.

Part I

Communicable diseases and epidemiology

1. Background and importance of modeling the spread of epidemics

1.1 Background of modeling the spread of epidemics

Modern approaches to epidemiological analysis and disease modeling emerged in the late 19th and early 20th centuries. One of the earliest breakthroughs involved mapping cases of cholera to identify patterns of transmission, leading to the hypothesis that contaminated water played a central role in spreading the disease. Around the same period, researchers also began to recognize and model the cyclic behavior of infectious diseases like measles and cholera using discrete-time models. These initial efforts in spatial and temporal epidemic analysis, combined with advancements in biology, contributed significantly to understanding how diseases spread. Beyond the development of vaccines and treatments, infectious disease research aims to mitigate health threats by uncovering the patterns of disease dynamics across time and space. To support this goal, numerous analytical and modeling methods have been created, grounded in the idea that disease transmission is influenced by underlying spatial structures inherent in both human and physical geography.[1]

Early disease modeling efforts primarily focused on mathematical representations at the population level, often relying on assumptions of homogeneity. These models typically divided the host population into distinct compartments, with individuals assumed to interact only within their immediate surroundings.

One of the most basic and widely known of these models is the SIR model, which was originally developed for closed populations. While such approaches have been valuable for estimating the scale of outbreaks, they often overlook the complex spatial and temporal dynamics that influence disease transmission across different communities factors that are critical for informing effective public health responses. Traditional non-spatial, population-based models do not explicitly account for the underlying causes of epidemic spread, limiting their ability to fully capture the mechanisms driving disease emergence and propagation.[2]

The advancement of computer technology and the growing availability of spatially referenced disease data have paved the way for more sophisticated modeling approaches. These tools enable the simulation of large populations and allow researchers to examine the progression of disease over time and across geographic space. Motivated by the need for more realistic representations, modern modeling efforts have evolved to include advanced mathematical frameworks, individual-based statistical models, and simulation-based techniques. Unlike traditional models, individual-based approaches explicitly account for key factors influencing disease transmission such as individual behavior, person-to-person interactions, and contact networks allowing them to more accurately reflect the heterogeneity observed in real-world outbreaks. These models also integrate data on host locations and movement patterns with detailed descriptions of infection processes and disease progression, making them powerful tools for analyzing observed epidemiological patterns and assessing potential intervention strategies. As a result, a wide range of spatial modeling approaches have been developed, encompassing both population-level and individual-level perspectives to better understand and predict the spread of infectious diseases. [3]

1.2 The importance of modeling the spread of epidemics

Epidemiological modeling plays a crucial role in managing public health crises by enabling experts to anticipate the course of an epidemic, including when it may peak and how many individuals could be affected. This foresight allows authorities to better prepare for potential impacts. Models also support the strategic allocation of limited resources such as medical supplies, hospital capacity, and healthcare personnel by identifying where and when interventions are most needed.

Through simulations, the potential outcomes of public health measures like lockdowns, vaccination campaigns, or social distancing can be evaluated before they are implemented, minimizing risk and maximizing effectiveness. Moreover, modeling offers a scientific foundation for policy decisions, helping to balance health priorities with economic and social considerations. It enhances our understanding of disease dynamics, shedding light on critical factors such as transmission rates, immunity development, and the roles different population groups play in spreading infections. On a global scale, models can also identify regions at higher risk of future outbreaks, enabling proactive responses from international health organizations. Overall, they serve as vital tools for evaluating and refining policies throughout the course of an epidemic.[4]

2. The impact of epidemics on public health and crisis management systems

2.1. The impact on public health

Pandemics cause significant health and economic challenges, especially in low- and middle-income countries, which often face a greater burden. Beyond the direct health impact, they disrupt economic activity through fear-driven behavior and restrictive measures like quarantines. These disruptions reduce productivity and can lead to long-term economic setbacks. In politically unstable regions, pandemic responses may also heighten tensions, potentially leading to conflict and weakening trust between governments and citizens.[5]

2.2. The impact on crisis management system

Epidemics severely challenge crisis management systems by overwhelming healthcare infrastructures, disrupting economies, and testing public trust. They require rapid coordination of resources, clear communication to counter misinformation, and support for mental health and social stability. Economic disruptions, including job losses and business closures, demand emergency aid and recovery planning. Effective responses depend on collaboration across governments, international organizations, and the private sector. Epidemics also expose weaknesses in preparedness, emphasizing the need for stronger health systems, surveillance, and long-term resilience strategies.[6]

3. Challenges of classical models in the face of dynamic epidemics

Classical epidemiological models like the SIR framework face limitations in capturing the complex dynamics of real-world epidemics. They often assume a homogeneous population and fixed parameters, overlooking important factors such as individual behavior, geographic variability, and the emergence of new pathogen strains. These models also tend to focus on short-term outcomes, ignoring long-term effects like herd immunity or disease resurgence. As a result, more advanced approaches—such as agent-based, network, and stochastic models—are needed to better reflect the evolving nature of epidemics.[7]

Part II

Epidemic modeling

Mathematical modeling

1. Presentation of Classical Epidemiological Models SIR, SEIR, and their variants

The SIR and SEIR models are foundational frameworks in epidemiology used to analyze and forecast the spread of infectious diseases within a population. The SIR model categorizes individuals into three groups: *Susceptible* (S), who are at risk of infection; *Infected* (I), who can transmit the disease; and *Recovered* (R), who have gained immunity after infection. This model operates under the assumption of homogeneous mixing, where every individual has an equal

probability of contact with others, and it considers a closed population with no births or unrelated deaths. Once individuals recover, they are assumed to be immune to future infections.

The disease dynamics are modeled using a set of ordinary differential equations (ODEs) that describe the rate of change in each compartment:

Where β is the transmission rate, γ is the recovery rate, and (N) is the total population. The basic reproduction number (R). Defined as:

$$\begin{aligned}\frac{dS}{dt} &= -\beta \frac{SI}{N} \\ \frac{dI}{dt} &= \beta \frac{SI}{N} - \gamma I \\ \frac{dR}{dt} &= \gamma I \\ \frac{dR}{dt} &= \gamma I(t)\end{aligned}$$

The basic reproduction number, R_0 , is defined as the average number of secondary infections produced by a single infectious individual in a fully susceptible population and is given by:

$$R_0 = \frac{\beta}{\gamma}$$

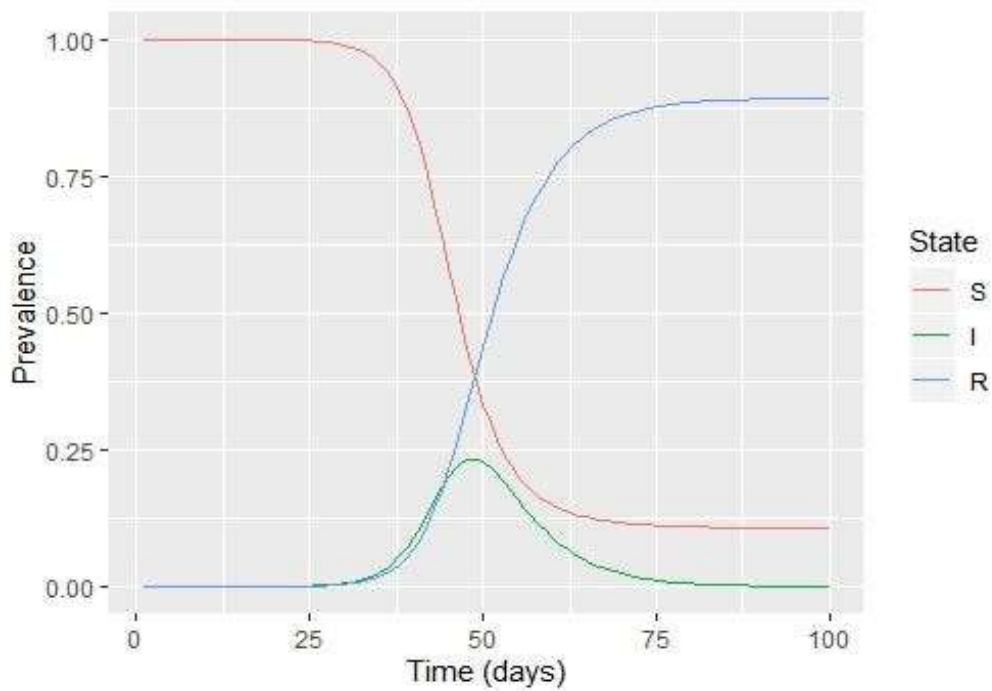


Figure 1: Growth of infection and depletion of the susceptible population in an SIR outbreak

Plays a critical role in determining the potential for disease spread. If ($R_0 > 1$), the disease will spread; if ($R_0 < 1$), the disease will eventually die out.

The SEIR model extends the SIR model by adding an *Exposed (E)* compartment, which represents individuals who have been infected but are not yet infectious due to an incubation period. This model is particularly useful for diseases with a delayed onset of infectiousness, such as COVID-19. The SEIR model consists of four compartments:

- Susceptible (S)
- Exposed (E)
- Infected (I)
- Recovered (R)

The dynamics are governed by the following equations:

$$\frac{dS}{dt} = -\beta S(t)I(t)$$

$$\frac{dI}{dt} = \beta S(t)I(t) - \gamma I(t)$$

Various extensions of the classic SIR and SEIR models have been developed to better reflect real-world disease dynamics. These include models accounting for waning immunity (SIRS and SEIRS), incorporation of vaccination effects (SIV), age-specific differences in transmission and recovery, and network-based models that consider heterogeneous contact patterns within populations. Such adaptations allow for more accurate and nuanced predictions of disease spread and control measures.

2. Differential equations used, assumptions, and conditions of application

The SIR and SEIR models are mathematical models used to describe the spread of infectious diseases within a population. These models rely on differential equations to represent the rate of change in different compartments of the population. Let's explore the differential equations used, along with assumptions and conditions of application.

2.1. SIR Model (Susceptible, Infected, Recovered)

The SIR model divides the population into three compartments:

- S: Susceptible individuals (those who are at risk of being infected)
- I: Infected individuals (those who are currently infected and can transmit the disease).
- R: Recovered individuals (those who have recovered and are assumed to have immunity).

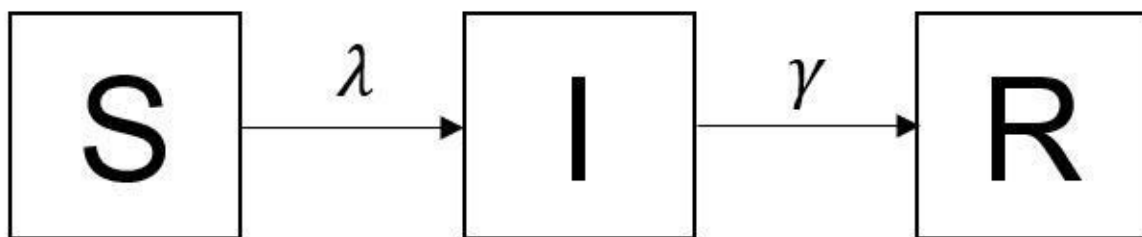


Figure 2: transmission diagram for the SIR model

The basic differential equations governing the SIR model are:

$$\frac{dS}{dt} = -\beta SI$$

This represents the rate at which susceptible individuals get infected. It depends on the transmission rate (β), the number of susceptible individuals (S), and the number of infected individuals (I).

$$\frac{dI}{dt} = \beta SI - \gamma I$$

This equation represents the rate of change in the infected population. The first term (βSI) reflects the number of new infections, while the second term (γI) represents the rate of recovery, where γ is the recovery rate.

$$\frac{dR}{dt} = \gamma I$$

This describes the rate of change in the recovered population, with γ representing the rate at which infected individuals recover.

Assumptions

- The total population remains constant, so:

$$N = S(t) + I(t) + R(t) .$$

- Individuals are assumed to either be susceptible, infected, or recovered. No individuals can be reinfected or return to a susceptible state (no immunity loss).
- The disease spreads through direct contact between susceptible and infected individuals, and recovery occurs at a constant rate.
- The transmission rate β and recovery rate γ are constant over time.

Conditions of application

- This model assumes no birth or death rates except for those caused by infection (no external or background mortality).
- The disease has a finite duration of infectivity (people eventually recover).
- It is applied when there is no need to account for more complicated epidemiological factors, such as external interventions or varying susceptibility.

2.2. SEIR Model (Susceptible, Exposed, Infected, Recovered)

The SEIR model is a more advanced extension of the SIR model that incorporates an "Exposed" (E) compartment, representing individuals who have been infected but are not yet infectious during a latent period.[8]

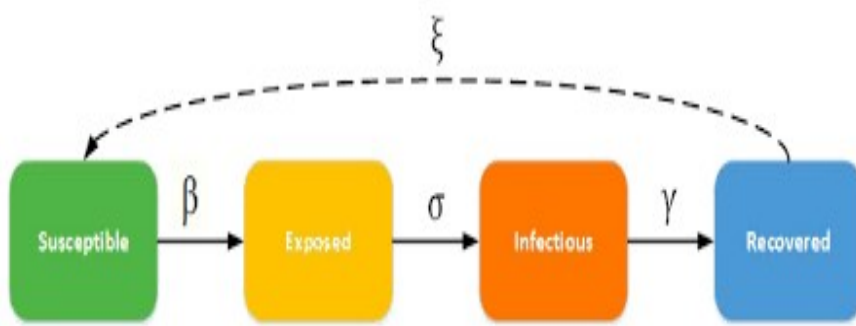


Figure 3: transmission diagram for the SEIR model

The **SEIR** model divides the population into four compartments:

- S: Susceptible individuals
- E: Exposed individuals (those who have been exposed to the virus but are not yet infectious)
- I: Infected individuals (those who are infectious)
- R: Recovered individuals

The differential equations governing the SEIR model are:

$$\frac{dS}{dt} = -\beta SI$$

As in the **SIR** model, this equation represents the rate at which susceptible individuals become exposed.

$$\frac{dE}{dt} = \beta SI - \sigma E$$

This equation represents the rate at which susceptible individuals become exposed (βSI) and the rate at which exposed individuals become infectious (σE). Here, σ is the rate at which exposed individuals progress to the infectious state.

$$\frac{dI}{dt} = \sigma E - \gamma I$$

This equation represents the progression of exposed individuals to the infected state and the recovery of infected individuals at the rate γ .

$$\frac{dR}{dt} = \gamma I$$

The same as in the SIR model, this describes the rate at which infected individuals recover.

Assumptions

- $S + E + I + R = N$, where N is the total population.
- The latent period is assumed to be constant for all exposed individuals.
- The transmission rate β , the rate at which exposed individuals become infectious σ , and the recovery rate γ are constant.
- The disease spreads through direct contact between susceptible and infected individuals.
- Individuals are assumed to either be susceptible, exposed, infected, or recovered, with no reinfection and no immunity loss.

Conditions of Application

- This model is more suitable for diseases with a significant latent period between exposure and becoming infectious (e.g., diseases like COVID-19 and tuberculosis).
- It is appropriate when the exposure period needs to be accounted for separately from the infectious period.
- Similar to the SIR model, it assumes constant-rates and does not typically include external factors like vaccination, varying population sizes, or varying contact rates.

2.3. Key Differences between SIR and SEIR Models

- **SIR model:** Assumes individuals become infectious immediately after being infected.
- **SEIR model:** Accounts for an incubation period where individuals are exposed but not yet infectious.

3. Practical Applications of the Models and Analysis of Their Limitations

3.1. Practical application to the SIR model

The SIR (Susceptible–Infected–Recovered) model serves as a valuable tool for understanding and managing the spread of infectious diseases in real-world settings. Public health officials use it to estimate key parameters such as the basic reproduction number R_0 , predict the peak of an epidemic, and determine how long an outbreak might last. By simulating various scenarios, the model helps evaluate the potential impact of intervention strategies like vaccination, quarantine, or social distancing. For example, during the COVID-19 pandemic, the SIR model was used to forecast case numbers, assess healthcare system capacity, and inform decisions on

lockdown policies. Its simplicity makes it especially useful in the early stages of an outbreak when data is limited, offering quick insights to guide public health responses.

Epidemiological models such as SIR and SEIR have found wide-ranging applications beyond traditional disease modeling. In health, they help analyze influenza outbreaks—like one in a British boarding school where SIR dynamics revealed a basic reproduction number (R_0) of 3.652 and are used to understand vector-borne diseases like dengue through coupled human-mosquito models, or to address emerging infectious diseases such as SARS, incorporating super-spreader effects. In social networks, these models describe the spread of rumors, user behavior, viral marketing, and even audience applause by simulating contagion-like dynamics of influence. In informatics, they aid in studying the dissemination of files in peer-to-peer networks and the propagation of computer viruses, taking into account their stealthy and destructive nature. In economics and finance, SIR-like models inform rational expectations in public health decision-making and model financial contagion in interbank systems, illustrating how shocks can cascade across interconnected institutions and nations. Across all these domains, the models provide valuable insights into how entities are they viruses, ideas, or financial crises—spread through populations or networks. [9]

3.2. Practical application to the SEIR model

The SEIR model has been extensively used to analyze and predict the dynamics of infectious disease outbreaks, particularly those with an incubation or latent period. A notable example is its application during the COVID-19 pandemic. Governments and public health agencies around the world used SEIR-based simulations to forecast the spread of the virus, estimate peak infection periods, and evaluate the impact of interventions such as lockdowns, social distancing, and vaccination. By incorporating real-time data (such as confirmed cases and recovery rates), the SEIR model helped policymakers anticipate healthcare demand, allocate medical resources, and plan timely mitigation strategies. For instance, during the early phases of COVID-19, the SEIR model allowed researchers to estimate how long it would take for the virus to spread in a given population and what percentage of individuals might be infected if no measures were taken. This insight was crucial for designing responsive and targeted public health policies.

3.3. The limitations of the models

The SIR (Susceptible-Infected-Recovered) and SEIR (Susceptible-Exposed-Infected-Recovered) models, while widely used for modeling infectious diseases, have several limitations. They assume a homogeneous population, where everyone has the same likelihood of interacting with each other, neglecting the variation in social networks, behaviors, and movement patterns that exist in reality.

Furthermore, these models generally omit birth and death rates unrelated to the disease and do not account for external factors such as other diseases or environmental influences. The transmission dynamics are oversimplified, assuming that the infection spreads uniformly through contacts, with constant transmission probabilities, ignoring factors like seasonality, vaccination, or changes in behavior.

Hybrid models and the role of AI in modeling

1. Introduction to Hybrid models

Hybrid System-It is the combination of multiple approaches or techniques to solve problem. In the realm AI and ML, it often integrates both traditional Machine learning methods and techniques. In hybrid system is one that combines two or more distinct sub-system, often strengths of each to achieve optimal performance or functionality [10].

- **Structure:** A hybrid system comprises two or more distinct subsystems that interact and collaborate to achieve a common goal. These subsystems can operate on different principles, data representations, or computational paradigms.
- **Common Subsystems:** Symbolic/Rule-based Systems: Employ domain knowledge encoded in logical expressions or decision trees. Excellent for interpretability and handling structured data.
- **Statistical/Probabilistic Models:** Analyse data using statistical techniques like Bayesian networks or linear regression. Good for uncertainty quantification and reasoning under noise.

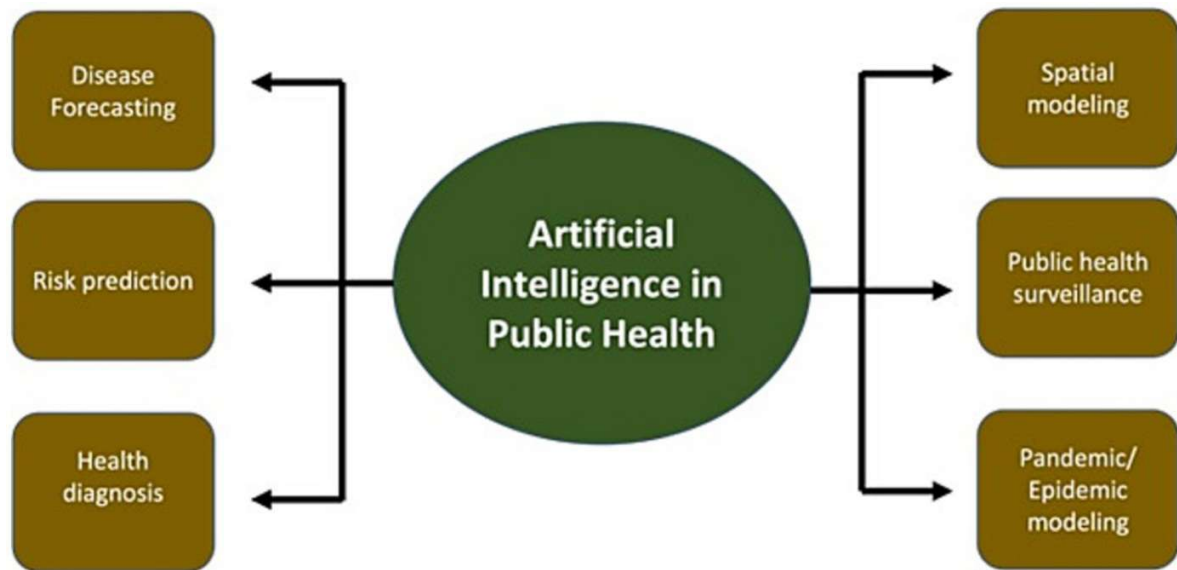


Figure 4: Applications of artificial intelligence in public health systems

2. An overview of artificial intelligence in epidemiology

Artificial Intelligence (AI) has become a powerful tool in epidemiology, particularly for forecasting the spread of infectious diseases. Various types of AI models are employed to predict, monitor, and manage disease outbreaks, each offering unique strengths. Machine Learning (ML) models are among the most commonly used, particularly supervised learning techniques like linear regression, which predict continuous outcomes such as the number of disease cases over time, and random forests, an ensemble method that aggregates decision trees to improve accuracy. Support vector machines (SVM) are used to classify data and handle high-dimensional datasets, making them useful for distinguishing between different disease outcomes.

In cases where labeled data is unavailable, unsupervised learning methods, such as clustering algorithms (e.g., K-means or DBSCAN), help identify hidden patterns in large datasets, such as locating disease hotspots or detecting unusual patterns in disease spread [11].

Deep Learning models, such as neural networks, are particularly well-suited for handling large, complex datasets, including electronic health records, mobility data, and environmental factors. Recurrent neural networks (RNNs) and Long Short-Term Memory (LSTM) networks, which are specialized in analyzing time-series data, have proven effective for forecasting future

disease cases by capturing temporal dependencies in data, such as predicting the trajectory of disease outbreaks based on past trends.

Convolutional neural networks (CNNs), though originally used for image processing, are now applied to analyze geographical data and detect disease patterns from satellite imagery or heat maps, aiding in real-time monitoring and detection. Agent-Based Models (ABMs) simulate individual interactions within a population, offering a more dynamic and granular approach to disease spread. These models model individual behaviors and social networks, allowing the simulation of various policy interventions, such as vaccination campaigns or changes in mobility patterns, and their impact on disease transmission. Bayesian models, such as Bayesian networks and Markov Chain Monte Carlo (MCMC) methods, use probabilistic reasoning to account for uncertainty in disease transmission dynamics. They enable researchers to estimate the likelihood of different epidemic scenarios, adjust predictions based on new evidence, and model complex dependencies between variables such as weather, human behavior, and healthcare responses.

The applications of AI in epidemiology are vast. AI-driven models are used for disease prediction and surveillance, analyzing historical trends and real-time data to predict future outbreaks, especially in the early stages of new or re-emerging diseases. Real-time forecasting allows for continuous monitoring of disease spread, with up-to-date predictions that help health authorities make timely decisions. Outbreak detection is greatly accelerated by AI, which can identify irregularities in disease patterns that may indicate the onset of an outbreak. Additionally, resource allocation models optimize the distribution of critical resources like vaccines, medical staff, and hospital beds, ensuring that interventions are deployed effectively and efficiently based on predicted disease trends. Ultimately.

3. Comparison of classical approaches and AI-based approaches

3.1. Accuracy and Efficiency: Comparing Detection Rates and False Positives/Negatives

Method	Accuracy	False Positives/Negatives
Traditional Methods		
Signature-Based Detection	High accuracy for known malware; struggles with zero-day threats and polymorphic malware.	Low false positives for known threats; high false negatives for new or modified malware.
Heuristic Analysis	Better for unknown malware via behavior pattern analysis; accuracy varies with rules and may need manual tuning.	Higher false positives due to broad rules; may flag legitimate software as suspicious.
Behavioral Monitoring	Effective for real-time malware detection; accuracy depends on behavior complexity; may miss subtle threats.	Can generate false positives if legit programs behave like malware; false negatives if malware uses evasion techniques.
AI-Based Methods		
Machine Learning	Supervised learning offers high accuracy with good data; unsupervised methods can detect novel threats; accuracy depends on data quality and training.	Lower false positives/negatives than traditional methods when well-trained; performance varies with data/model quality.
Deep Learning	High accuracy for complex malware patterns; effective for classifying diverse and new threats due to learning from large datasets.	Can significantly reduce false positives/negatives by capturing complex patterns; may still face issues with evolving or obfuscated malware.

Table 1: comparison of malware detection methods: accuracy and false positives/negatives

3.2. Computational and Data Needs for Traditional vs. AI Approaches

Method	Computational Needs	Data Needs
Traditional Methods		
Signature-Based Detection	Low computational overhead; relies on simple pattern matching.	Needs an up-to-date signature database; does not require large data volumes for each analysis.
Heuristic Analysis	Moderate computational needs for rule evaluation and data analysis.	Requires historical data to develop and refine rules; does not need large-scale real-time data.
Behavioral Monitoring	High computational requirements due to real-time monitoring, data logging, and complex analysis.	Needs substantial data from monitored systems to effectively detect behavior-based anomalies.
AI-Based Methods		
Machine Learning	Significant resources required for training large datasets; inference is typically less demanding.	Requires large volumes of labeled data for training; continuous data collection is important for maintaining accuracy.
Deep Learning	Very high computational demands for both training and inference; depends heavily on hardware (e.g., GPUs) due to complex neural network structures.	Requires extensive, diverse datasets to train models effectively and capture a wide range of malware behaviors.

Table 2: computational and data requirements of malware detection methods

AI-based methods offer superior accuracy and adaptability compared to traditional approaches, especially for detecting novel and sophisticated threats. However, they come with higher computational and data requirements. Traditional methods, while generally faster and less resource-intensive, struggle with new threats and may generate more false positives/negatives. Combining both approaches can provide a more robust and comprehensive malware analysis strategy. [12]

Conclusion:

Mathematical modeling plays a crucial role in understanding and predicting the dynamics of epidemic spread. Through models such as SIR, SEIR, and their numerous extensions, researchers can simulate transmission patterns, evaluate intervention strategies, and estimate key epidemiological parameters. These models serve not only as analytical tools for theoretical exploration but also as practical frameworks to guide public health decision-making. Despite their simplifications and assumptions, mathematical models provide valuable insights into complex biological and social processes underlying disease propagation. As data availability and computational power continue to grow, integrating models with real-time data and machine learning techniques will further enhance their predictive accuracy and applicability in managing future outbreaks.

Chapter II

IA approach, Deep-learning, LSTM and BiLSTM

II. IA approach and Deep-learning

The objective of artificial intelligence (AI) research is to develop computer systems capable of performing tasks using thinking processes similar to those of human beings. Thus, the challenge lies not only in understanding human thought, but also in its modelling and reproduction.

Artificial intelligence (AI) has become a topic of great importance both in the media and in scientific journals, mainly due to the many achievements that have resulted from it, many of which are the result of advances in machine learning. AI is based on a learning approach to replicate, through an application, system or process, part of human intelligence. Examples of artificial intelligence systems include facial recognition and visual perception. Machine learning (ML), a subdomain of AI, uses artificial neural networks (ANN) to mimic how humans make decisions. Machine Learning allows computers to develop learning models autonomously, without prior programming, based on vast datasets. Just below is Deep Learning (DL), one of the many approaches to machine learning that has been hugely successful in recent years. The DL is characterized by deep neural networks capable of performing complex tasks more efficiently and accurately.

In this chapter, we will discuss the different forms of machine learning as well as the different types of neural networks. We will begin with a general overview of these concepts, and then focus on artificial neural networks (ANN), recurrent neural networks (RNN) and convolutional neural networks (CNN). [13]

1. Definition of machine learning (Machine Learning)

Machine learning, as a subdomain of artificial intelligence (AI), focuses on creating systems that can learn and improve their performance based on the data they analyze. Artificial intelligence and machine learning are often mentioned together, highlighting the close relationship between these two areas [14].

2. Type of learning

Deep learning algorithms can be divided into two main categories: supervised and unsupervised learning:

- Supervised learning uses a tagged set of learning data, which means that each sample data is associated with a label that indicates its class or expected value. There are two main types of supervised learning models: regression models, which aim to predict continuous values, and classification models, which seek to assign examples to predefined discrete categories.
- Unsupervised learning, on the other hand, is learning models from unlabeled data. In this case, the algorithm must automatically extract structures or groupings in the data without receiving explicit information about the expected categories. Unsupervised learning is about discovering intrinsic relationships and organizing data according to their similarities or underlying motives.

These two types of supervised and unsupervised learning play important roles in the field of deep learning, offering complementary approaches to extract useful information and perform advanced data analysis tasks.

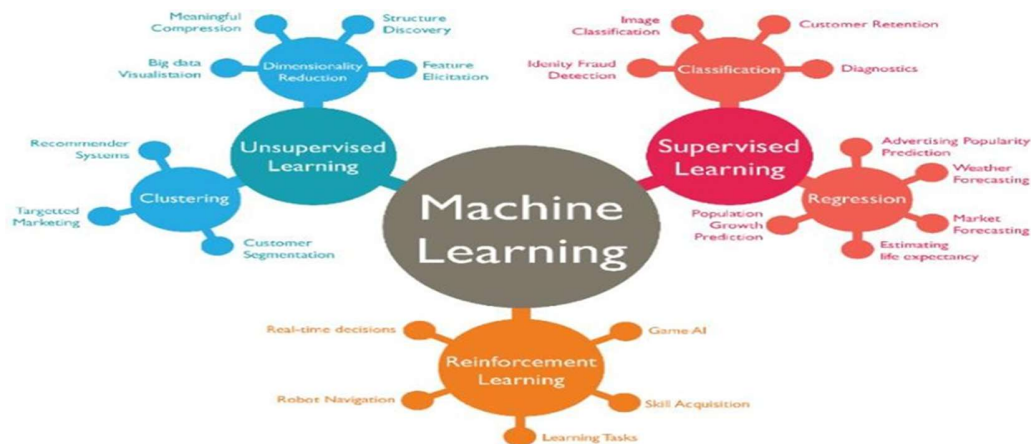


Figure 5: Diagram of the different use cases for a given type of training

3. Deep Learning Definition (DL)

Deep learning (DL) is a class of machine learning (ML) in which information is processed using hierarchical layers to understand data representations and characteristics at increasing levels of complexity. Deep learning is also known as hierarchical or structured deep learning.

All deep learning algorithms are based on neural networks (NRT) that share some fundamental properties. These networks are composed of interconnected neurons and organized in layers. What distinguishes them is the specific network architecture that is how neurons are organized, and sometimes the method used for their learning.

Deep learning has led to significant advances in areas such as computer vision, speech recognition, machine translation and many other areas of AI. The power of deep neural networks makes it possible to extract complex representations and perform sophisticated tasks by exploiting hierarchical relationships and latent data characteristics.

4. Neural Networks

4.1. Biological neuron

Many inventions have been inspired by nature, and artificial neural networks (ANN) are no exception. They are inspired by the architecture of the human brain and its biological neurons. The human brain is made up of about 10 billion neurons, and each neuron is connected to about 10,000 other neurons. Biological neurons are composed of cellular bodies containing a nucleus, as well as many branched extensions called dendrites. They also have a long extension called axon. At the end of the axon are tiny structures called axon endings, which are connected to dendrites in other neurons (see Figure 6) [15].

This complex organization of the human brain has been a source of inspiration for the development of artificial neural networks, which attempt to reproduce these connections and neuronal interactions to perform learning and information processing tasks.

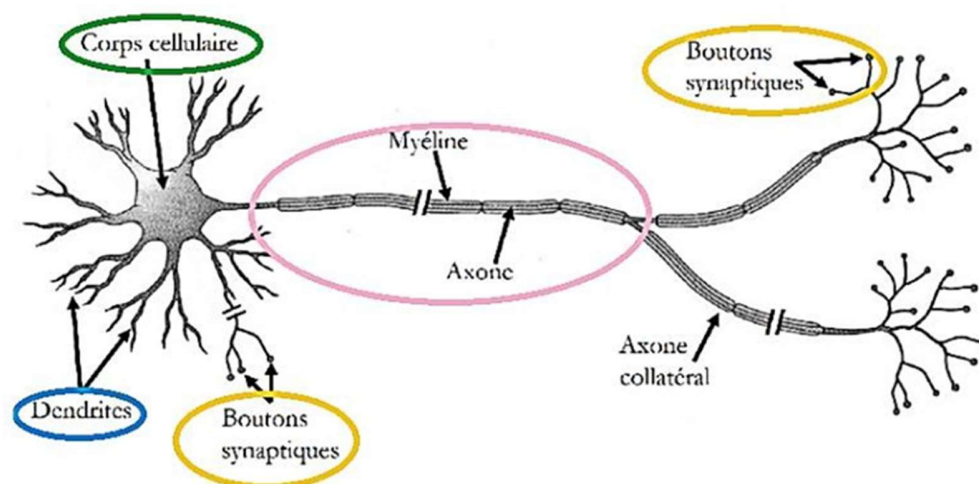


Figure 6: Structure of a biological neuron

Each neuron receives electrochemical signals from other neurons via its dendrites. If these inbound signals are strong enough to activate the neuron, it is stimulated and transmits the signal along its axon, which then relays it to the dendrites of other neurons. These neurons in turn can be activated, thus prolonging the message transmission process.

Artificial neural networks seek to mimic this process by simulating interactions between neurons and signal propagation.

4.2. Principle of neural networks (Neural Network)

A neural network is a powerful tool for analyzing the complex relationships between input and output data. It is based on the principle of learning, where the network adjusts its internal parameters, called synaptic weights, in order to capture the models and knowledge present in the data. By learning and optimizing these synaptic weights, the network can generalize and make accurate predictions on new data. By combining many interconnected neurons and using learning techniques, neural networks are able to model complex relationships and provide effective solutions to a variety of problems.

4.3. Functioning of Neural Networks

The functioning of neural networks is based on the distribution of variable values in units called neurons. These neurons are responsible for combining the information they receive from each other in order to determine the value of the discrimination coefficient. This ability to discern depends on the connectivity between neurons, that is, the links between units responsible for integrating information. Each neuron receives numerical information from its neighboring neurons, which are weighted by weights representing the strength of the connection. Calculations are then made in each neuron, and the result of this process is transmitted to the neurons downstream.

Neural networks use mathematical operations and activation functions to process and analyze data.

5. Types of Neural Networks

There are several architectures of neural networks commonly used in practice. Some of the most popular architectures are:

- Artificial neural networks (ANN)
- Recurrent neural networks (RNN)
- Convolution neural networks (CNN)

5.1. Artificial Neural Networks (ANN)

The forward neural network (feedforward) is one of the simplest forms of ANN. In this type of network, the data or input moves in a single direction, passing through the input nodes to reach the output nodes. It can be composed of hidden layers or without hidden layer. The network simply propagates data linearly across the network, without feedback, usually using an activation function to introduce non-linearity into the model [16]. This type of network is often used for classification and prediction tasks, where input data is transformed into a corresponding output using the weights and activation functions of the network.

5.1.1. Artificial neural network architecture

Neural networks are sophisticated structures of artificial neurons that receive multiple inputs and generate one or more corresponding outputs (Figure 7).

It consists of:

- Input Layer: is the set of neurons that carries the network input signal, and subsequently all the neurons in this layer are connected to the next layer.
- Hidden Layer: they can be one or more, this is where the relationships between variables will be highlighted. Choosing the number of layers and neurons is intuitive and requires experience from the expert.
- Output layer: represents the result of the neural network, called prediction.

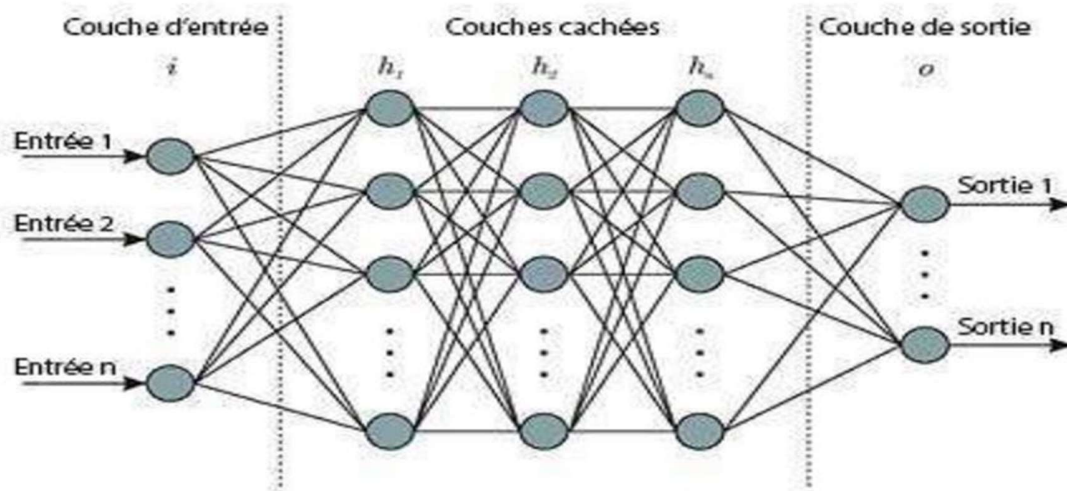


Figure 7: The architecture of an artificial neural network

5.1.2. Perceptron

The perceptron is an essential component of artificial neural networks, representing an artificial neuron model capable of performing a non-linear algebraic function. It acts as a receiver for one or more inputs, marked x_j and each input is individually weighted by w_j . The w_j weights determine the relative importance of the different inputs for the final output of the perceptron. The weighted sum of the inputs is then added to the b_j neuron bias and used as an argument for an activation function f . The use of an activation function introduces non-linearity into the model, allowing the perceptron to learn complex relationships between inputs and outputs.

When it comes to predicting with a neural network, the data is entered as inputs into the first layer, and the calculations propagate through all successive layers until they reach the final outputs. This process is commonly referred to as forward propagation or direct pr

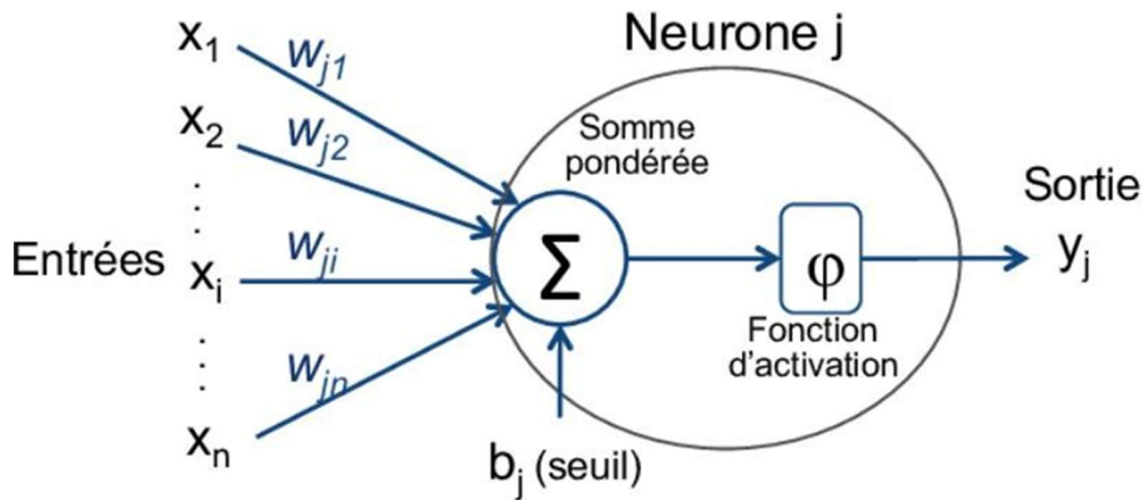


Figure 8 : La structure d'un neurone artificiel

5.2. Recurrent Neural Networks (RNN)

Unlike forward-propagated neural networks (Feed-Forward Neural Network -FFNN), the recurrent neural networks (Recurrent Neural Network - RNN) are characterized by the presence of at least one feedback loop in their connection graph. Over the past 30 years, several types of RNN have been developed, such as Elman networks, Jordan networks and Echo State Networks. Recently, one type of RNN has become widely used due to its outstanding performance in various tasks: Long-Short-Term Memory (LSTM - Long Short-Term Memory) neural networks. LSTMs play a central role in this thesis. [17]

In this section, we look at the simplest version of RNN proposed by Jeff Elman in 1990. In this approach, links are added to a multi-layer perceptron (Multi- Layer Perceptron - MLP), so that one layer of the network receives not only the current output from the previous layer, but also its own output at the preceding time step. This change has a considerable impact when inputs are in the form of temporal or spatial sequences, unlike the case where x input vectors are independent of each other.

Indeed, the structure of RNN introduces a memory mechanism of previous inputs that persists in internal states of the network and can influence all future outputs. In theory, this simple modification makes it possible to approximate with arbitrary precision any function that transforms an input sequence into a given output sequence, unlike the MLP.

However, it is important to note that RNNs of this type can be particularly difficult to drive (although recent developments have improved this situation), and without special precautions, the context usable is generally limited.

These types of RNN include:

5.2.1. Hopfield Networks

Hopfield networks are a type of recurrent neural network used in unsupervised learning applications and optimization problem solving. The network functions as a nonlinear associative memory, capable of retrieving and recognizing patterns stored in a data space. The main idea behind Hopfield networks is to create attractors in the network state space. They are widely used for tasks such as memory retrieval, pattern classification, optimization problem solving and dynamic system modeling.

5.2.2. LSTM

Short-term long-term memory (LSTM) cells were proposed by Hochreiter and Schmidhuber in 1997. They were developed to solve the problem of learning long-term dependencies in traditional recurrent neural networks (RNN).

LSTM (Long Short Term Memory) is one of the recurrent architectures used in artificial neural networks. LSTMs are composed of layers of neurons that are connected recursively, allowing the previous state of a neuron to be used as context to generate an output. LSTM cells are able to store information over longer time periods, allowing long-term dependencies to be captured in the data sequences. LSTMs have been very successful in many applications, including machine translation, text generation and speech recognition.

The effective formation of recurrent neural networks (RNN) is one of the main challenges they face. The main problem is the propagation of the gradient through time steps, which leads to stability and convergence problems during learning. To solve this challenge, several techniques have been developed, such as adaptive activation function, RNN variants and advanced optimization techniques. These advances make it possible to effectively train NRNs and successfully apply them to a wide variety of tasks.

5.2.3. GRU

The GRU (Gated Recurrent Unit) network is a network of recurrent neurons (RNN) composed of input, hidden and output layers. The input layer consists of neuronal segments determined by the space dimension of the input characteristics. The output layer is determined by the size of the desired output space. GRU cells are designed to regulate the flow of information and control the omission or addition of new information to the network's internal memory. GRU cells have gates (gates) that control access to memory and update its contents. The GRU network is widely used in applications such as machine translation, text generation, and speech recognition.

5.2.4. JANET

The Gated Recurrent Unit (GRU) is a recurring network variant that offers a simplified architecture by removing some ports from the LSTM. This has led to a new type of network called JANET, which retains only the memory port and cell memory, resulting in a more general model that is less demanding in computing power. This simplification of the architecture provides practical benefits and facilitates the application of these models in different tasks. [18]

The Gated Recurrent Unit (GRU) is a simplification of the LSTM because it does not use explicit cell states. The GRU uses a single reset port to control the amount of changed information

in the state cache, making it simpler and more effective in certain situations. This simplifies the LSTM architecture by avoiding the use of explicit cell states.

5.3. Convolution Neural Networks (CNN)

Convolutional neural networks are designed for signal and image processing, and have been widely used in the field of computer vision. Convolution layers capture local patterns in the input data, allowing important features to be extracted and highlighted.

ConvNets have been widely used in areas such as signal processing and image classification, and are highly effective in object detection, facial recognition, image segmentation, image classification, etc. Open CV provides advanced tools for image processing and computer vision. Convolutional neural networks (CNNs) are particularly effective for image processing because they can detect features at any location.

Learning CNNs involves optimizing parameters, including convolution cores, to minimize the difference between network outputs and corresponding "ground truth" tags. The use of CNNs is widespread in various fields, including computer vision, object recognition, medicine, automotive and mobile applications.

5.3.1. CNN Diapers

CNN is a mathematical construction that generally consists of three types of distinct layers (or building blocks): convolution, pooling and fully connected layers.

- Fully connected the last layer of a deep network, the classical perception layer, operates the final discrimination between images to be recognized, for example. The previous layers build and extract their characteristics.
- Convolution uses a dimension reduction to create a convolution on the input signal.
- Pooling (POOL) dimension reduction by substituting a part of the entries (sub-image) with a value, usually the maximum.

5.3.2. The backpropagation algorithm

The backpropagation learning algorithm starts by circulating forward the input data until it gets an input computed by the network. Then, the calculated output is compared to the known actual output. The weights are modified so that the error made between the calculated outputs is minimized at the next iteration. Taking into account the presence of hidden layers, the error is back propagated backwards to the input layer while changing the weighting. All examples are repeated until an output error is considered negligible. [19]

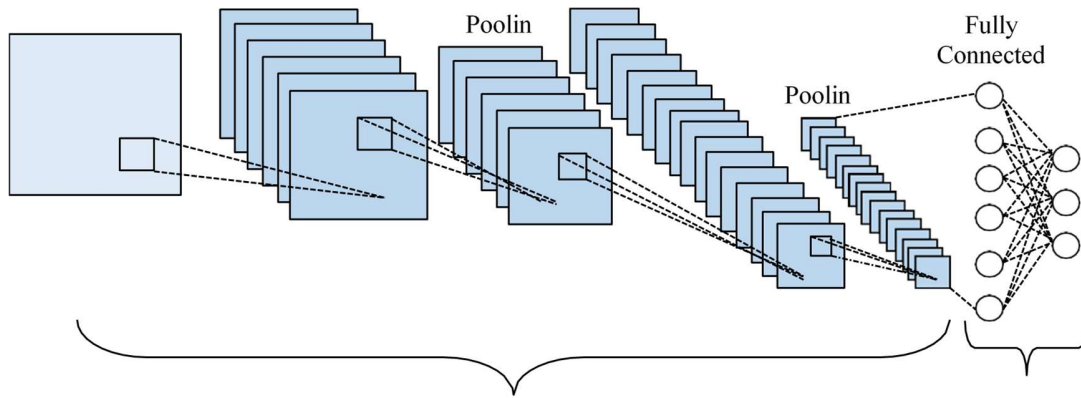


Figure 9: CNN's diagram

II. LSTM and BiLSTM

1. LSTM

The types of recurrent neural networks (RNN) that have been developed to model and process sequential data are LSTM (Long Short-Term Memory), BiLSTM (Bidirectional LSTM) and GRU. They were all intended to solve the problem of addiction and forgetting information in traditional NDR.

Special LSTM doors allow them to access and store information for long periods of time. In sequential modelling, BiLSTM can take into account past and future context through their unidirectional and bidirectional connections. On the other hand, GRU use a simpler method with fewer doors, which makes them faster in training and execution.

In this chapter, we present different metrics: the average, the maximum value (max), the minimum cost... etc.

The BILSTM and GRU models and the ADAM and SGD optimization techniques are used to optimize neural network models, including the LSTM and GRU architectures.

Finally we will highlight the evaluation of hyper-parameters such as learning rate, mini-batch size, number of times and number of hidden layers, is essential to achieve optimal performance

1.1. Metrics

- La formule moyenne générale est exprimée mathématiquement en:

$$\text{Average} = \frac{\text{Sum of Observations}}{\text{Total number of observations}}$$

- Determine the maximum value of the quadratic equation:

For example, it can be stated that if the equation has been represented as

$$ax^2 + bx + c$$

- The formula to find the maximum value will be:

$$\text{MAX} = c - \left(\frac{b^2}{4a} \right)$$

- The minimum cost using the equation:

$$\text{MIN} = c - \left(\frac{b^2}{4a} \right)$$

- RMS (Root Mean Square):

$$x_{RMS} = \sqrt{\frac{x_1^2 + x_2^2 + \dots + x_n^2}{n}}$$

When $\mathbf{x}$ = of data values.

And $\mathbf{n}$ = total number of items.

- Standard deviation:

$$\sigma = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n}}$$

- Variance is calculated as:

$$s^2 = \frac{\sum (x_i - \bar{x})^2}{n-1}$$

- Kurtosis formula:

$$K = \frac{\sum (x_i - \bar{x})^4}{ns^4}$$

- Peak-to-Peak value:

$$P = \frac{MAX - MIN}{2}$$

- Peak factor equation:

$$PI = \left(\frac{MAX}{MIN}\right)^2$$

- Peak factor equation:

$$M = \text{Mean} (\text{abs}(x))$$

- Impulse Factor:

$$Fi = \frac{P}{M}$$

- Clearance Factor:

$$Fc = \left(\frac{P}{M}\right)^2$$

1.2. Definition of Long Short-Term Memory (LSTM)

Hochreiter and Schmidhuber (1997) introduced short-term long-term memory cells (LSTMs) to learn long-term dependencies [20].

The most common recurrent architectures in artificial neural networks are LSTM (long-term memory). Like other recurrent networks, it consists of layers of connected neurons.

LSTMs can solve the gradient problem and manage long-term dependencies, which distinguishes them from other recurring architectures. When learning deep neural networks, LSTMs avoid the problem of gradient disappearance or explosion.

Information is stored in cells or memory blocks in LSTMs. The internal state of the network is maintained and updated by these LSTM computing units. This allows LSTMs to store information over long time sequences and make predictions based on previous information.

The use of doors, such as oblivion, entry and exit doors, which regulate the flow of information through the cell is an important feature of LSTMs. These doors allow LSTMs to control the information that is transmitted, forgotten or stored on exit [21].

Memory blocks are a set of repeatedly connected blocks that make up an LSTM layer. These blocks can be considered as different memory chips in a digital computer. Each year one or more memory cells. Repeatedly connected and equipped with three multiplicative units - the input, output and oblivion gates - that provide continuous analogues for writing, reading and resetting operations for cells [22].

1.3. The architecture of an LSTM model

1.3.1. LSTM Weights

A memory cell in an LSTM architecture has weight parameters for input, output and internal state. These parameters allow to weight the different information and to build the internal state of the cell according to the steps of the input time.

- Input weights are used to weight the current time step entry. They determine the relative importance of each element of the entry for the memory cell.
- Output weights are used to weight the output of the last time step. They control the influence of the previous output on the current output of the memory cell.
- The internal state is an intermediate state that is used in the calculation of the output for the current time step. It is built according to current input, previous internal state and input weights. This internal state allows the memory cell to store information over long time sequences and maintain a long-term memory [23].

1.3.2. LSTM Doors (Gated)

An LSTM cell consists of three main doors: the Forget Gate, the Input Gate and the Output Gate. Each door is responsible for regulating the flow of information through the cell.

- The forgetting door determines which information is important to forget from the cell. It takes into account the previous hidden state and the current entry to decide which information is relevant and should be retained.

- The front door controls new information to be added to the hidden state. It looks at the current entry and the previous hidden state to determine what new information should be included.
- The exit door decides which part of the hidden state should be exposed to the LSTM's exit. This output can be used to make a prediction or as an input for the next steps in the model.

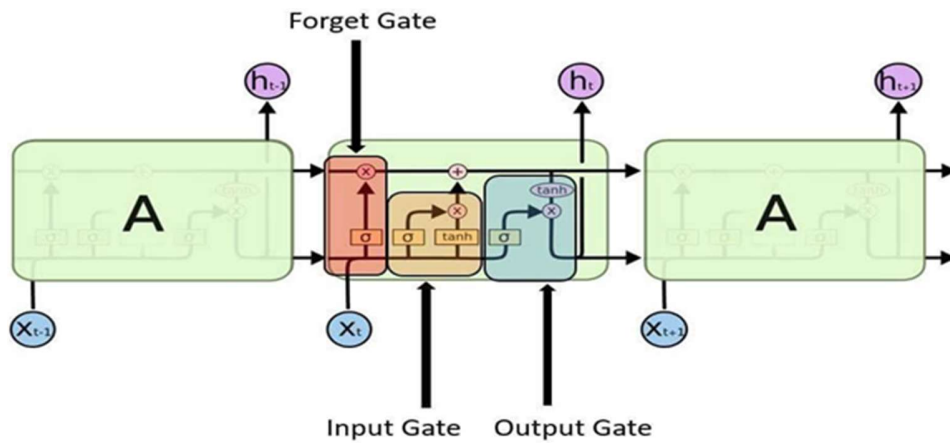


Figure 10: LSTM doors

The main structure of LSTM networks consists of an input layer, one or more hidden layers and an output slice. Let $\{x_1, x_2, \dots, x_t\}$ the input sequence of an LSTM, where x_t Represents a vector introducing the dimension of the feature to the time step t -th. The number of neurons in the input layer is equal to the number in the function space. The most important component of the LSTM structure is the memory cell. The state of the memory cell C_{t-1} interacts with the intermediate h_{t-1} output and the input to update the internal state vector according to the outputs, the previous time step and the inputs of the current time phase. In addition, the memory cell also defines an input node, an input port, and an output port to maintain and adjust the state of this cell. The calculation process is described in equations.

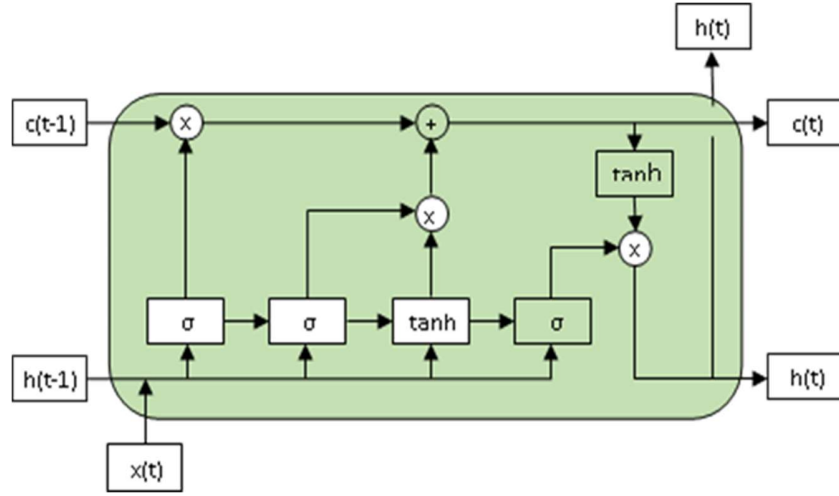


Figure 11: LSTM architecture.

$$f_t = (W_f x_t + V_f h_{t-1} + b_f)$$

$$i_t = (W_i x_t + V_i h_{t-1} + b_i)$$

$$C_t = \tanh(W_c x_t + V_c h_{t-1} + b_c)$$

$$C_t = f_t * C_{t-1} + i_t * C_t$$

$$o_t = (W_o x_t + V_o h_{t-1} + b_o)$$

$$h_t = o_t * \tanh(C_t)$$

$x_t \in R^d$ and $h_t \in R^k$ are the inputs and outputs of the LSTM unit. $i_t \in R^k$, $f_t \in R^k$, $o_t \in R^k$ vectors activation of the front door, forget the door and exit door, respectively. All parameters $W \in R^{k \times d}$ $V \in R^{k \times d}$ $b \in R^k$, have parameters that can be learned and it is the multiplication by element by, σ represents the sigmoid activation function.

2. Bidirectional LSTM Model

The basic LSTM network considers only the previous context of the data without future context. Given that time series sensor data has strong temporal dependencies, processing of this data is insufficient. In order to design the time information of future and past contexts, a bidirectional LSTM structure is built. This structure processes the sequence data in both the front and back paths with two hidden layers, then the forward and backward time information will be introduced into the same output layer. The structure of the BiLSTM network is illustrated in Figure 12.

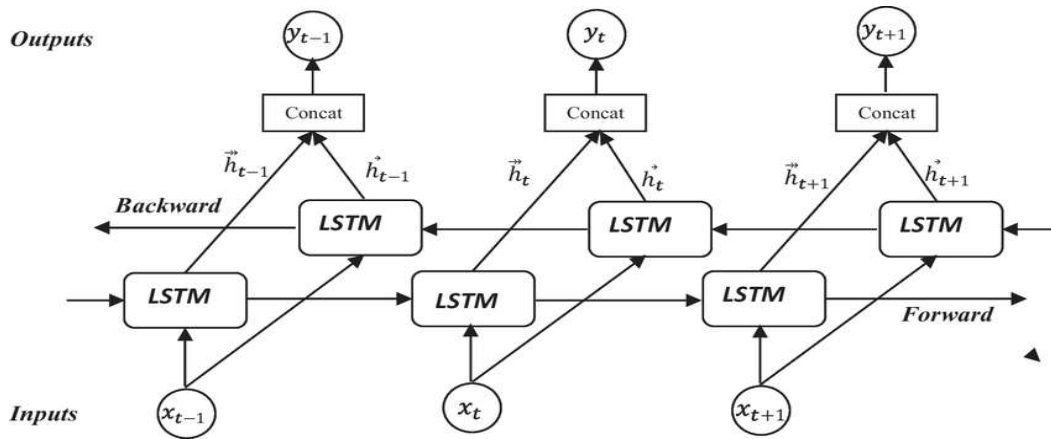


Figure 12: Basic structure of a BiLSTM model

2.1. What is Bi-LSTM and how it works:

To understand Bi-LSTM, let's break down its components and functionality:

- **LSTM (Long Short-Term Memory):** LSTM is a type of RNN designed to overcome the limitations of traditional RNNs in capturing long-term dependencies in sequential data. It introduces memory cells and gating mechanisms to selectively retain and forget information over time. LSTMs have an internal memory state that can store information for long durations, allowing them to capture dependencies that may span across many time steps.

- **Bidirectional Processing:** Unlike traditional RNNs that process input sequences in only one direction (either forward or backward), Bi-LSTM processes the sequence in both directions simultaneously. It consists of two LSTM layers: one processing the sequence in the forward direction and the other in the backward direction. Each layer maintains its own hidden states and memory cells.
- **Forward Pass:** During the forward pass, the input sequence is fed into the forward LSTM layer from the first time step to the last. At each time step, the forward LSTM computes its hidden state and updates its memory cell based on the current input and the previous hidden state and memory cell.
- **Backward Pass:** Simultaneously, the input sequence is also fed into the backward LSTM layer in reverse order, from the last time step to the first. Similar to the forward pass, the backward LSTM computes its hidden state and updates its memory cell based on the current input and the previous hidden state and memory cell.
- **Combining forward and Backward States:** Once the forward and backward passes are complete, the hidden states from both LSTM layers are combined at each time step. This combination can be as simple as concatenating the hidden states or applying some other transformation.

The benefit of Bi-LSTM is that it captures not only the context that comes before a specific time step (as in traditional RNNs) but also the context that follows. By considering both past and future information, Bi-LSTM can capture richer dependencies in the input sequence.

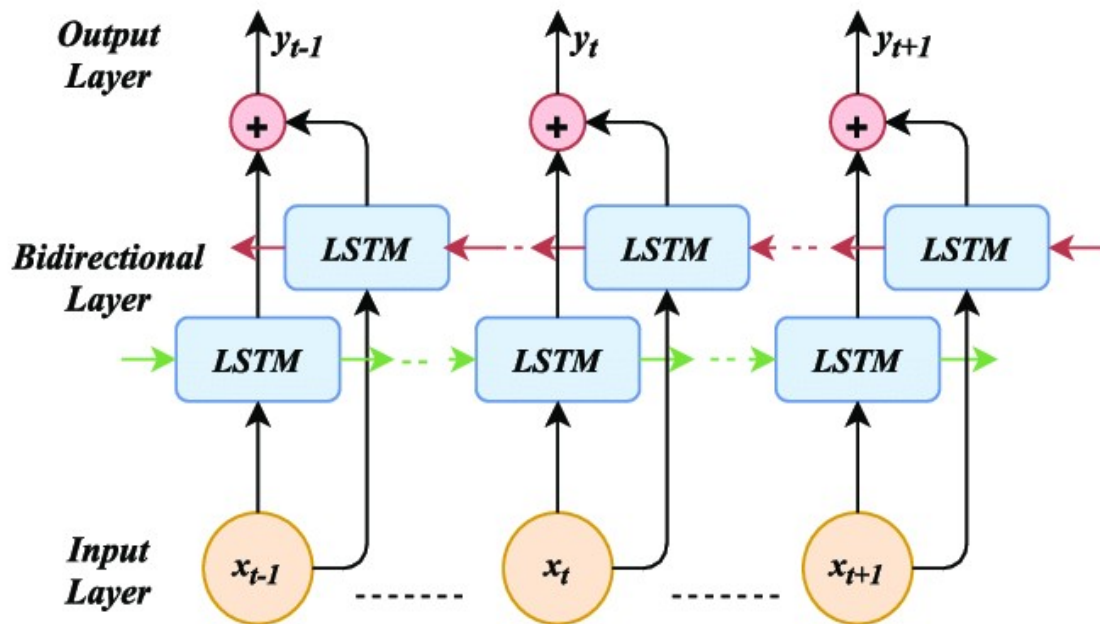


Figure 13: BiLSTM model showing the input and output layers

2.2. BiLSTM architecture

Let's break down each component of the architecture

- **Input Sequence:** The input sequence is a sequence of data points, such as words in a sentence or characters in a text. Each data point is typically represented as a vector or embedded representation.
- **Embedding:** The input sequence is often transformed into dense vector representations called embedding. Embedding capture the semantic meaning of the data points and provide a more compact and meaningful representation for the subsequent layers.
- **Bi-LSTM:** The Bi-LSTM layer is the core component of the architecture. It consists of two LSTM layers: one processing the input sequence in the forward direction and the other in the backward direction. Each LSTM layer has its own set of settings.
- **Output:** The output of the Bi-LSTM layer is the combination of the hidden states from both the forward and backward LSTM layers at each time step. The specific combination method can vary, such as concatenating the hidden states or applying a different transformation.

The Bi-LSTM layer processes the input sequence in both forward and backward directions simultaneously. During the forward pass, the LSTM layer captures information from the past (previous time steps), while during the backward pass, it captures information from the future (following time steps). This bidirectional processing allows the model to effectively capture long-term dependencies in the input sequence.

The output of the Bi-LSTM layer can be used for various purposes depending on the specific task. For example, in text classification, the output may be passed through a fully connected layer followed by a soft-max activation to obtain class probabilities. In sequence labeling tasks like named entity recognition, the output may be directly used to predict the label for each input token.

The architecture of a Bi-LSTM can be further extended or modified based on the specific requirements of the task. Additional layers, such as fully connected layers or attention mechanisms, can be added on top of the Bi-LSTM layer to further enhance the model's capabilities and performance.

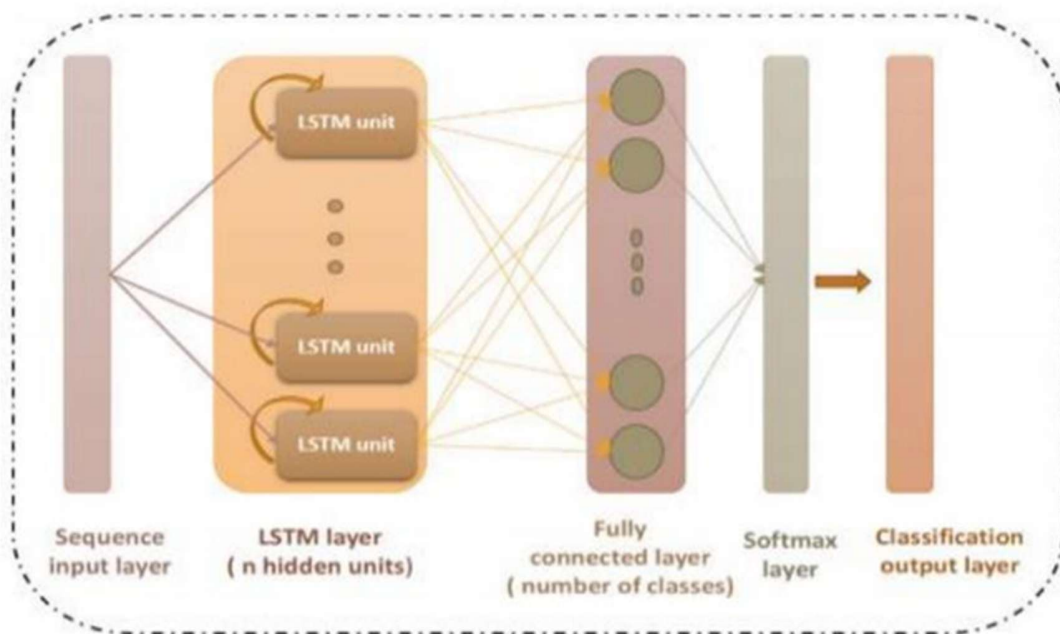


Figure 14: BiLSTM architecture

Conclusion

The use of CNNs and RNNs was an optimal solution given the results obtained in several research projects.

This chapter is a state of the art on concepts related to the issue addressed in this thesis and its design. It has been divided into two parts. In the first, we quickly went over some definitions and clarifications regarding smart home. In the second chapter, we approached deep learning with some detail, to show its architectures that we have chosen for our application.

In the following chapter, we review the work carried out in the context of the problem addressed which is the prediction of energy consumption in a smart home.

Chapter III

Modeling Approach

1. Introduction

In this chapter, two different approaches will be examined for modeling data related to COVID-19—an enhanced SIR model and a deep-learning BiLSTM model. In this enhancement, the SIR model goes beyond the classical epidemiological framework by introducing infection and recovery rates that vary with time in a manner that reflects real-world conditions, wherein public health interventions and improved treatment change these parameters over time. The LSTM approach, on the other hand, is data-driven and capable of identifying more complex patterns in epidemiological time series without the need for any explicit mathematical formulation of the disease dynamics. Therefore, looking at both mechanistic and machine learning approaches will help us put together some complementary knowledge of the pandemic progression and lay down the foundation for more trustworthy epidemic forecasting.

2. Dataset

The dataset serves as a chronicle of COVID-19 cases reported from different countries, spanning across the globe. It has slightly over 32,000 entries, with each entry detailing critical data points on confirmed cases, deaths, recoveries, and active cases. The data is organized on a day-by-day basis starting on January 22, 2020, allowing for an in-depth chronological tracking of the evolution of the pandemic in different countries. For every country covered in the dataset, the information captures how the situation unfolded over time, reflecting the varying severity and timelines of COVID-19 outbreaks.

This extensive collection tracks not only the cumulative counts but also reflects the dynamic changes in the status of the pandemic concerning active cases; the mortality rates into their recovery rates vary from period to period. The dataset provides individual nations with the knowledge base to assess their responses to the crisis, how transmission patterns have changed over time across countries, and how public health interventions may or may not have shaped the pandemic.

Having an all-embracing data set from almost every corner of the world and over all time frames, it becomes a vital resource for researchers and analysts and policymakers who are trying to investigate trends, compare outbreaks from region to region, gauge the success of their mitigation strategies, or model future behavior of infectious diseases. Such depth and width make it strong to perform many analyses related to COVID.

3. SIR model

SIR (Susceptible-Infected-Removed) model which was developed by Ronald Ross, William Hamer, and others in early 20th century. It consists of a system of three coupled, non-linear ordinary differential equations, which does not possess an explicit formula solution. It is however straightforward to extract a great deal of information about the solutions using simple tools from calculus. The following simple model shows us how it is used in laying theoretical foundation for public health interventions, including several cornerstones that required such a model to discover [24].

3.1. Initial Conditions

For the SIR model to function effectively, it is essential to set the initial conditions for the three primary compartments: susceptible, infected, and recovered individuals. These initial values represent the state of the population at the onset of the simulation, and they directly influence how the disease spreads throughout the population. There are two primary ways to determine these initial values: through user inputs or by extracting values from the latest available data.

- **User Input:** When it comes to inputting their own initial conditions into the model, users are not disappointed. This becomes a direct means of customizing the simulation to each user's individual scenario or dataset. They can even begin by directly entering completely real observation numbers, such as the number of infected, recovered, and susceptible individuals, and make it more realistic to users, thus improving accuracy.

- **Default Extraction from Latest Data:** If the user does not directly input the initial conditions for the run, the model will automatically extract them from the most recent available data. This data-driven approach makes sure that the model is reflective of the present state of the epidemic, without requiring the manual input of values from the user.

3.1.1. Loss Function and Optimization

A critical step in the training and optimization of the enhanced SIR model involves comparison of predictions made by the model with real-world data- in this case, smoothed infection and recovery data. This step allows for model parameter adjustment and refinement to achieve better accuracy. For this purpose, we designed a weighted mean squared error (MSE) loss function that gives a quantitative measure of the deviation of the model's predictions against the real observed data. The optimization of the model parameters is expected to minimize this loss function.

3.1.2. Weighted Mean Squared Error (MSE) Loss Function

- The mean squared error or MSE loss function is the standard way to calculate the difference between the predicted and actual values in regression problems. For our model in this context, it will represent the square of differences between the predicted numbers of infected individuals I and recovered individuals R , and the actual data points [25].

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_{pred}(t_i) - y_{true}(t_i))^2$$

- Thus, our model incorporates a weighted form of MSE that pays more attention to the more current data because more recent incidence and recovery trends should be deemed the most relevant for understanding the current dynamics as well as future patterns of the disease. In other words, it ensures more weightage to recent changes in infection and recovery rates, which is significant while predicting future scenarios [26].

$$\text{Weight MSE} = \frac{1}{n} \sum_{i=1}^n w_i \cdot (y_{pred} - y_{true}(t_i))^2$$

Where w is the weight assigned to each data point, which can vary with.

Specifically, we assign higher weights to recent data points by using an exponentially decaying weight function:

$$w = \frac{\exp\left(\frac{i}{n}\right)}{\exp(1)}$$

Here (i) represents the index of the data point, and n is the total number of data points. This formula ensures that the most recent points are given more influence in the loss calculation, allowing the model to better reflect the latest trends in infection and recovery.

By giving more weight on recent data, the model here is thus able to react faster to changes in the epidemic's journey, which is valuable in unpredictable and fast-changing scenarios such as the COVID-19 period.

3.1.3. Optimization of Model Parameters

The subsequent significant phase in the training of SIR models is the optimization of parameters, which implies finding the optimal values of the model parameters, resulting in the minimization of loss function [27]. These parameters include:

- $\beta_{\max}$: The maximum infection rate, representing the highest possible rate of infection.
- $\beta_{\min}$: The minimum infection rate, reflecting how interventions (such as social distancing or vaccination) reduce transmission over time.
- $\gamma_{\max}$: The maximum recovery rate, representing the highest possible rate of recovery.
- $\gamma_{\min}$: The minimum recovery rate, which accounts for the possibility that recovery rates might slow down over time.
- **Half-life**: A parameter that controls the rate at which the infection rate decreases and the recovery rate increases over time due to interventions (like vaccines or better treatment options).

Due to the complex, nonlinear nature of the model, the Differential Evolution algorithm was selected for optimization. This evolutionary algorithm is particularly useful where the model poses many parameters or is highly nonlinear, particularly because it does not require the model to be differentiable or even continuous.

3.1.4. Differential Evolution Algorithm

This technique is a global optimization one that mimics processes of natural evolution. Iteratively, it exposes a population of candidate solutions (sets of parameters) to the crossover, mutation, and selection operations for improvement. The basic steps of the Differential Evolution algorithm are:

- **Initialization:** A random population of parameter sets (candidate solutions) is generated.
- **Mutation:** For each individual in the population, new candidate solutions are created by combining the current individual with others in the population. The mutation introduces diversity into the population.
- **Crossover:** **The mutated solutions are combined with the original solutions to** create new individuals. This process promotes exploration of the parameter space.
- **Selection:** The best-performing solutions (those that minimize the loss function) are kept, and the rest are discarded.

By using Differential Evolution, we ensure that the algorithm effectively explores a wide range of possible parameter values and converges to the optimal set of parameters for the model.

3.2. Simulation and Prediction

After the optimization of model parameters, the next step involves simulating the dynamics of the SIR model to predict the progression of the disease over time.

The system of ordinary differential equations (ODEs) that defines the SIR model is solved numerically using the `solve_ivp` function from the SciPy library. This method provides a flexible and efficient way to integrate initial value problems for ODEs over a specified time interval [28].

The simulation is carried out using the optimized parameters obtained from the Differential Evolution algorithm, allowing us to forecast the evolution of the following compartments:

- **Susceptible (S):** These are those individuals who are not infected, however, could become infected. A susceptible individual may become infected or remain susceptible. As the virus spreads from its source or new sources occur, more individuals will become infected, thus the susceptible population will increase for a period of time (surge period)[29].

- **Infected (I):** These are those individuals who have already been infected by the virus and can transmit it to those individuals who are susceptible. An infected individual may remain infected, and can be removed from the infected population to recover or die [30].
- **Recovered (R):** These are those individuals who have recovered from the virus and are assumed to be immune, $R_m(t)$ or have died [31].

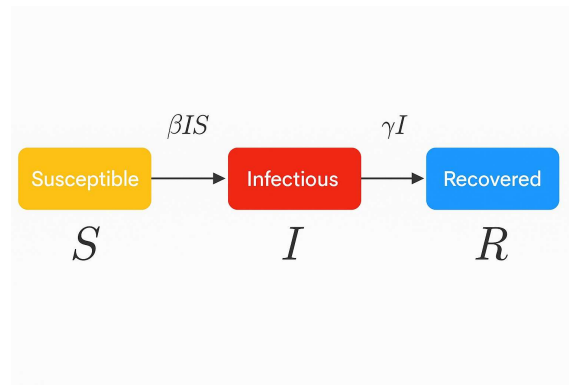


Figure 15: SIR model

The system evolves to the SIR model's differential equations:

$$\frac{ds}{dt} = -\beta \frac{SI}{N}$$

$$\frac{dI}{dt} = \beta \frac{SI}{N} - \gamma I$$

$$\frac{dR}{dt} = \gamma I$$

Where:

- $\beta(t)$ is the time-dependent infection rate.
- $\gamma(t)$ is the time-dependent recovery rate.
- N is the total population.

3.2.1. Comparison with Actual Data

After solving the model over the desired time horizon, the predicted values of infected and recovered individuals are compared against the actual reported epidemiological data.

To ensure a fair comparison:

- The model's outputs are smoothed to match the smoothing applied to the real-world data.
- Both predicted and observed curves are aligned based on initial conditions and timescales.

The quality of the model's predictions is evaluated using statistical measures, such as the R^2 score, which quantifies how well the predicted curve matches the actual observed curve. A higher R^2 value indicates a better fit between the model and reality.

By comparing the simulated dynamics with real data, we assess the effectiveness of the optimized model in capturing the trends of infection spread and recovery in the population.

3.3. Results and Analysis

Completing the optimization and simulation phases, the model's projections were compared to epidemiological data from the real world for performance evaluation and insight extraction.

3.3.1. Visualization of Predictions

The results are visualized by plotting both the actual reported data and the model's predicted values over time:

- Actual infections and recoveries are represented as transparent scatter points, reflecting the real- world recorded cases.
- Model predictions are shown as bold continuous lines, representing the expected evolution of infections and recoveries based on the SIR model.
- A shaded region around the prediction line, covering 10%, is included to illustrate a basic confidence interval and highlight the model's uncertainty margins.

This visualization allows a direct, intuitive comparison between the observed dynamics and the model's forecasts.

3.3.2. Model Evaluation: R^2 Score

The model's predictive accuracy is quantitatively assessed using the coefficient of determination (R^2 score) [32]. This metric evaluates how well the model predictions align with the actual data:

- **High R^2 values** (closer to 1) It is suggested that the model represents most of the variability observed in data [32].
- **Separate R^2 scores** The peak day values for infected and recovered individuals permit an estimation of the two epidemic trends CrossRef, thus forming a complementary outcome.[32].

An R^2 value close to 1 for both infected and recovered populations indicates a strong model fit and reliable simulation performance.

3.3.3. Key Extracted Metrics

In addition to prediction accuracy, important epidemiological parameters were extracted from the fitted model:

- **Initial and final infection rates** (β_0 and β_f): Reflect how transmission dynamics evolved due to interventions and behavioral changes.
- **Initial and final recovery rates** (γ_0 and γ_f): Indicate improvements in recovery over time, possibly due to medical advances.
- **Basic reproduction number** (R_0):
 - R_0 at the start of the simulation gives an estimate of the disease's contagiousness without intervention.
 - R_0 at the end reflects the impact of interventions and natural disease dynamics.
- **Estimated half-life of interventions**: Measures the speed at which interventions (e.g., vaccination, restrictions) effectively reduce transmission rates.

These metrics provide deep insights into the epidemic behavior and the effectiveness of control measures over time.

The enhanced SIR model provides a more realistic simulation of epidemic dynamics by allowing infection and recovery rates to vary over time. The use of weighted loss functions and differential evolution optimization improved the model's ability to match real-world data. Extracted epidemiological metrics give deeper insights into the effectiveness of public health measures and the natural course of the epidemic. This approach can be further extended to forecast future trends or evaluate different intervention strategies.

4. Deep learning model LSTM

4.1. Dataset and Feature Engineering

This model applies the same dataset on COVID-19, as earlier analyzed, encompassing daily records from various countries. However, compared to the earlier method, the deep learning approach incorporates a wider set of features in order to capture diverse dimensions of the pandemic's progress:

- **Core Features:** confirmed cases, deaths, recovered cases, and active cases
- **Rate of Change Features:** new confirmed cases, new deaths, and new recovered cases
- **Derived Metrics:** mortality rate and recovery rate.

The new feature set enables the model to learn from many indicators at once, thereby maximizing the possibility of capturing complex relationships that have not been modeled by conventional epidemiology approaches. Addition of first-order derivatives (i.e., daily changes) and proportional metrics enhances the model's capability of detecting trends and patterns in the data.

4.2. Data Preprocessing

4.2.1. Feature Scaling

Keeping in mind data normalization for each feature by means of the MinMaxScaler transformation applied to the training data before entering the neural network, this procedure ensures that all features are on a comparable scale (ranging from 0 to 1) which is an important aspect while training a neural network because:

- Prevents features with larger magnitudes from dominating the learning process
- Improves convergence speed during optimization.
- Enhances numerical stability throughout the training process.

4.2.2. Sequence Creation

The temporal nature of epidemiological data requires specialized handling [33]. The model transforms the time series data into supervised learning format through the following steps:

- **Sequence Formation:** For each time point, a sequence of previous observations (sequence length = 21 days) is assembled to serve as input features.
- **Multi-step Output:** Rather than predicting just one day ahead, the model is designed to forecast a full week (7 days) of active cases simultaneously.
- **Sliding Window Approach:** The sequences are created using a sliding window technique, where each sequence is offset by one day from the previous.

The chosen sequence length of 21 days (3 weeks) represents a balance between capturing sufficient historical context and maintaining a practical number of training samples. This window size is particularly suitable for COVID-19 data as it encompasses approximately two typical incubation periods [33].

$$X = \{x_{t-seqlength+1}, x_{t-seqlength+2}, \dots, x_t\}$$

$$y = \{y_{t+1}, y_{t+2}, \dots, y_{t+forecast_horizon}\}$$

Where

- X represents the input sequence of feature vectors
- y represents the target sequence of active cases
- $seq_length = 21$ (days)
- $forecast_horizon = 7$ (days)

4.3. Model Architecture:

4.3.1. Bidirectional LSTM Network

The core of the prediction system is a bidirectional LSTM (BiLSTM) neural network. Unlike standard LSTMs that only process sequences in forward direction, bidirectional LSTMs process data in both directions, offering several advantages:

- **Complete Temporal Context:** By processing sequences in both directions, the model can capture dependencies that might be more apparent when considering future-to-past relationships in addition to past-to-future ones.
- **Enhanced Feature Extraction:** BiLSTMs often extract more meaningful features from time series data by considering the full context surrounding each time point.

4.3.2. Layer Configuration

The model employs a carefully designed architecture with multiple specialized layers:

- **First BiLSTM Layer:** 128 units with sequence return, incorporating L2 regularization (0.001) to prevent overfitting
- **Dropout Layer (30%):** Randomly deactivates 30% of neurons during training to improve generalization
- **Second BiLSTM Layer:** 64 units without sequence return, creating a condensed representation
- **Dropout Layer (30%):** Additional regularization to prevent co-adaptation of features
- **Dense Layer:** 32 units with ReLU activation for non-linear feature transformation
- **Dropout Layer (20%):** Final regularization layer
- **Output Layer:** Dense layer with 7 units (one for each day in the forecast horizon) without activation function for regression output

This progressive narrowing of layer dimensions (from 128 to 64 to 32 to 7) creates a funnel-like architecture that gradually distills complex temporal patterns into focused predictions.

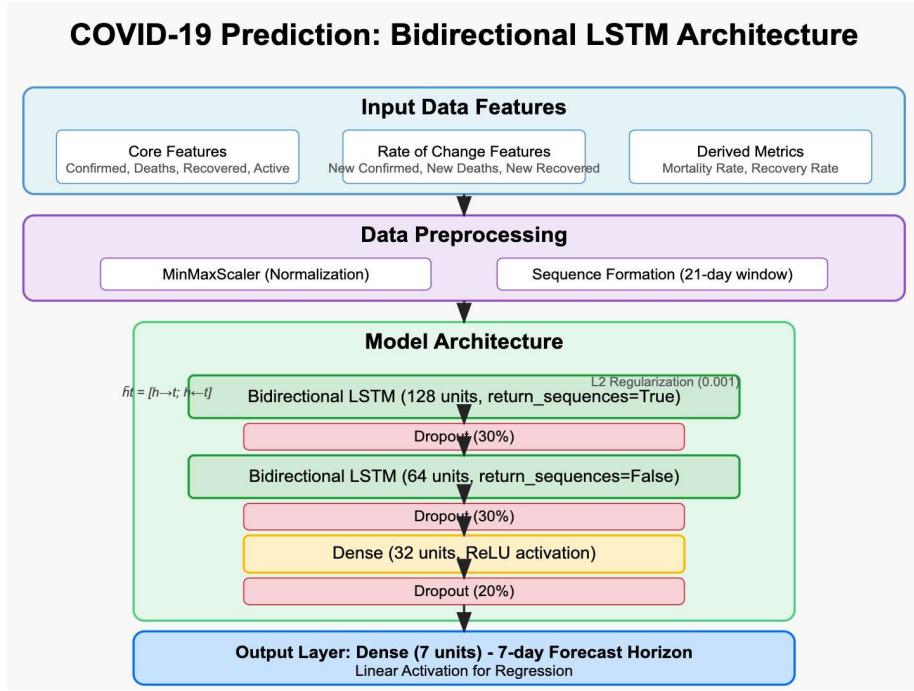


Figure 16: LSTM model architecture

The mathematical formulation of a BiLSTM cell can be expressed as:

$$\vec{h}_t = \overrightarrow{LSTM}(x_t, \vec{h}_{t-1})$$

$$\overleftarrow{h}_t = \overleftarrow{LSTM}(x_t, \overleftarrow{h}_{t+1})$$

$$h_t = [\vec{h}_t; \overleftarrow{h}_t]$$

Where:

$\vec{h}_t$ is the forward hidden state at time t

$\overleftarrow{h}_t$ is the backward hidden state at time t

h_t is the concatenated bidirectional state

4.4. Training Strategy

4.4.1. Optimization Algorithm

The Adam model optimizer has been developed based to bring the best of both worlds based on the incentives of AdaGrad and RMSProp optimizers. Due to Adam's capability of adaptive learning rate, it is most productive in deep learning applications dealing with large parameter spaces. The initial learning rate is 0.001, which balances the speed of convergence and stability during training [34].

The update rule for Adam optimization can be summarized as:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$$

$$\hat{m}_t = \frac{v_t}{1 - \beta_1^t}$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t}$$

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon} \hat{m}_t$$

Where:

- g_t is the gradient at time t
- m_t is the first moment estimate
- v_t is the second moment estimate
- β_1 and β_2 are decay rates for the moment estimates
- η is the learning rate
- ϵ is a small constant for numerical stability

4.4.2. Loss Function

The Mean Squared Error (MSE), which is the optimization objective, penalizes the squared difference between corresponding predicted and true values directly. The loss function suits regression problems and imposes a larger weight on larger errors, which helps in accurate predictions for significant outbreaks.

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_{pred,i} - y_{true,i})^2$$

4.4.3. Training Process Enhancement

Several techniques are employed to improve training efficiency and model performance:

- **Early Stopping:** Training automatically terminates if validation loss fails to improve for 15 consecutive epochs, preventing overfitting and unnecessary computation.
- **Learning Rate Reduction:** The learning rate is reduced by 50% after 5 epochs without improvement, allowing for more precise optimization as training progresses.

- **Best Model Preservation:** The model state with the lowest validation loss is automatically restored at the end of training.

The learning rate reduction follows the formula:

$$\eta_{new} = \eta_{old} \times factor \quad \text{When plateau is detected}$$

Where $factor = 0.5$ in this implementation.

4.4.4. Training Configuration

The model training process uses the following hyper parameters:

- **Epochs:** Maximum of 100 iterations through the entire dataset
- **Batch Size:** 32 samples processed before each parameter update
- **Validation Split:** 20% of data reserved for validation (performance monitoring)

The batch size of 32 represents a compromise between training speed and gradient estimate quality, while the generous maximum epoch allowance ensures the model has sufficient opportunity to converge, even with early stopping in place.

4.5. Model Evaluation

4.5.1. Performance Metrics

The model's performance is evaluated using multiple complementary metrics:

1. **Root Mean Square Error (RMSE):** Measures the standard deviation of prediction errors.

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_{pred,i} - y_{true,i})^2}$$

2. **Normalized RMSE:** Expresses the RMSE as a percentage of the maximum observed value, facilitating intuitive interpretation of error magnitude [35].

$$NRMSE = \frac{RMSE}{\max(y_{true})} \times 100\%$$

These metrics provide a general measure of accuracy, with the normalized metric giving an intuitive context of the magnitude of errors with respect to the scale of the data.

4.5.2. Visualization

Training progress is visualized through a learning curve plot that displays both training and validation loss across epochs. This visualization:

- Allows for monitoring of convergence behavior
- Identifies potential overfitting (divergence between training and validation curves)
- Highlights the epoch with minimum validation loss (optimal model state)

The use of logarithmic scale for the y-axis enables better visualization of loss improvements across orders of magnitude, making subtle improvements in later epochs more apparent.

5. Comparison with SIR Model

While the SIR model offers a mechanistic understanding of disease dynamics with interpretable parameters (β , γ , R_0), the deep learning approach provides complementary strengths:

- **Flexibility:** Automatically captures complex patterns without requiring explicit mathematical formulation
- **Feature Utilization:** Incorporates multiple data features beyond just case counts
- **Multi-step Forecasting:** Directly outputs predictions for multiple future time points simultaneously
- **Adaptability:** Can potentially adapt to changing dynamics without explicit re-parameterization

The deep learning model excels at capturing complex, potentially non-linear relationships in the data that might be difficult to express in closed-form equations. However, it sacrifices some interpretability compared to the SIR approach.

The key differences can be summarized in the following comparison table:

Aspect	SIR Model	Deep Learning Model
Mathematical Foundation	Differential equations	Neural network architecture
Interpretability	High (epidemiological parameters)	Lower (black-box approach)
Feature Requirements	Minimal (case counts)	Multiple features leveraged
Adaptation to Changes	Requires re-parameterization	Can learn changing patterns
Prediction Horizon	Flexible simulation	Fixed output window (7 days)
Computational Cost	Lower	Higher (training)

Table 3: Comparison between SIR and Deep Learning approaches

6. Conclusion

Our study has shown the importance of relying on both systematic and data-driven modeling approaches in studying COVID-19. An improved SIR model with time-dependent parameters can capture the time evolution of disease dynamics while maintaining interpretability via epidemiological measures like reproduction numbers. The BiLSTM deep learning method considers multiple features to generate accurate multi-step forecasts, thus bypassing the need for transmission dynamics to be mathematically formulated. Although the SIR model is more interpretable, deep learning better accounts for complicated, non-linear relationships present within the data. Hence, these two approaches are complementary and help in understanding pandemic pathways and determining the best interventions, thereby exemplifying the necessity for diversified modeling approaches in epidemiology and public health response.

Chapter IV

Application Architecture and Implementation

1. Introduction

The COVID-19 pandemic must be understood and anticipated as much as possible to make fair decisions. The prediction and analysis framework of the application is designed to hold a key to studying real-time data, producing forecasts, and comparing results through several models. By selecting a country, one can immediately access the country's key statistics, including total confirmed cases, total deaths, total recoveries, and active cases. These values are fetched and presented in interactive graphs that are intuitive and provide a snapshot of the current position.

2. Application Overview

The COVID-19 Analysis and Prediction Dashboard is an interactive application making extensive crawling and detailed analytics accessible to the end-user on demand. A user can study historical COVID-19 data for some countries to understand the past trends and present scenarios. With such a tool, trends can be drawn with cases, deaths, and recoveries through beautiful charts and diagrams. All kinds of forecasting can be done by exposing the user to advanced hybrid modeling of what could come next. Users will be able to tweak prediction parameters and visualization modes through the interface to give their analysis the look they want. All predictions can be downloaded for further analysis or reporting.

2.1. Sidebar Controls

The discontinuous panel contains all the controls to effectively manage analysis parameters. The dropdown lets a user zoom in on their country of interest. The display/input option for population data ensures the modeling considers demographic context. A date selection calendar allows users to specify time frames for their analysis. Custom SIR input parameter specifications allow for scenario testing and hypothesis approaches.

2.2. Data overview

Upon the application's launch, users are placed right into an intuitive interface that allows them to select a country of interest and enter data relevant for prediction in COVID-19. Once a country has already been selected, the application, in return, obtains its current population data so that the analysis can be tracked. It then accesses a large dataset of COVID-19 information and starts working on it. Based on all of this information, the application casts dynamic and interactive plots that display key stats: total confirmed COVID-19 cases, total deaths, total recoveries, and the number of active cases. This visualization goes a long way toward helping the users comprehend the impact of the pandemic within the entertained country. This population data, alongside current COVID statistics, ensures an Analyze trend, compare results from country to country, and thereby render their own decisions from the presented data. The application's purpose is to explain the COVID-19 overview crisply and depictively for every selected region (Figure 3).

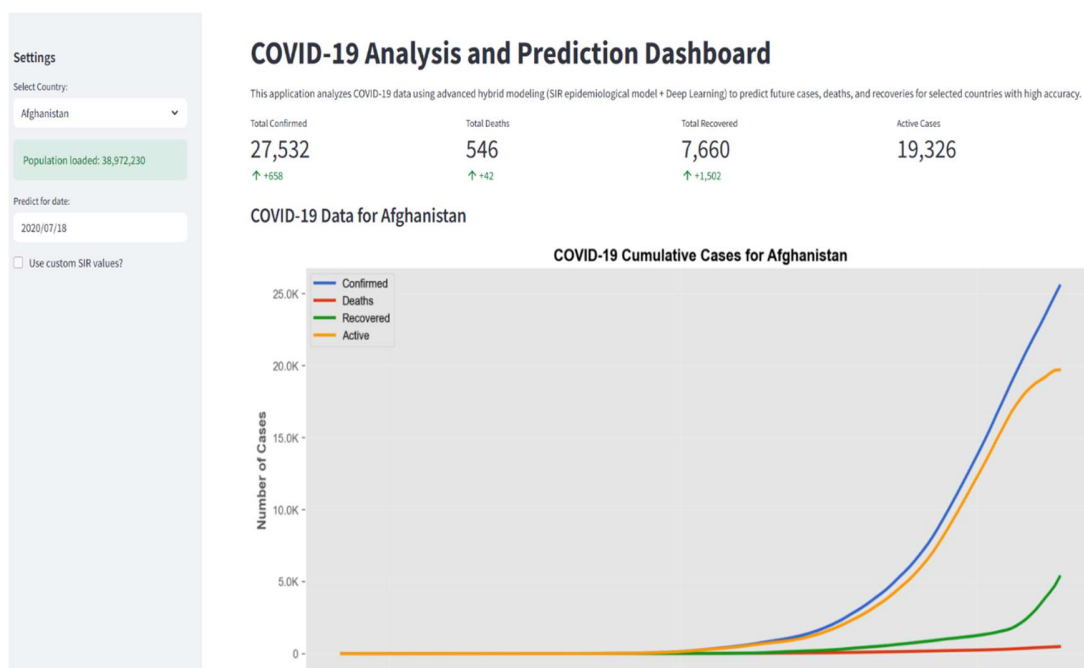


Figure 17: Data overview page

2.3. Get prediction

Moreover, scrolling down in the application brings about two buttons, each linked to a different prediction approach, letting users choose between the classical epidemiological model and a deep learning-based model. In detail:

- **SIR model button:** Clicking the SIR model button will set the application to make predictions using the Susceptible-Infected-Recovered SIR model. Once the computations are finished, a plot is displayed with the historical data and the predicted trend from a given date. Thus, the results are presented visually and numerically, providing an insight into how the virus is expected to spread (Figure 4).

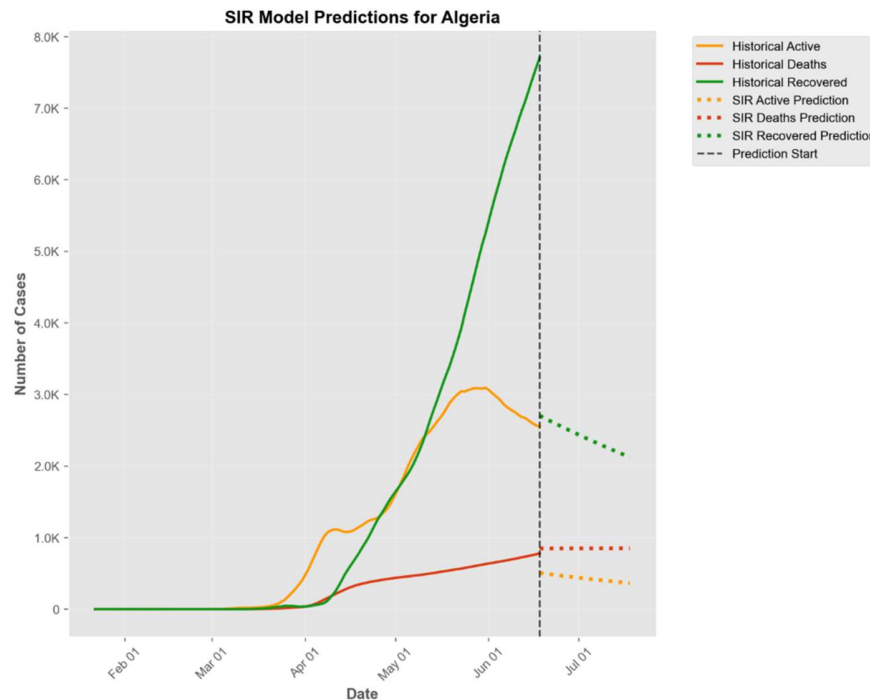


Figure 18: The SIR model prediction

- **Deep learning model button:** The deep learning button triggers the prediction using a trained Long Short-Term Memory neural network. Setting a country and specifying a date range in the sidebar, the model makes forecasts for active cases, deaths, and recoveries.

The predictions are then displayed as numerical values and plotted as time series for users to visualize the projected evolution of the pandemic (Figure 5).

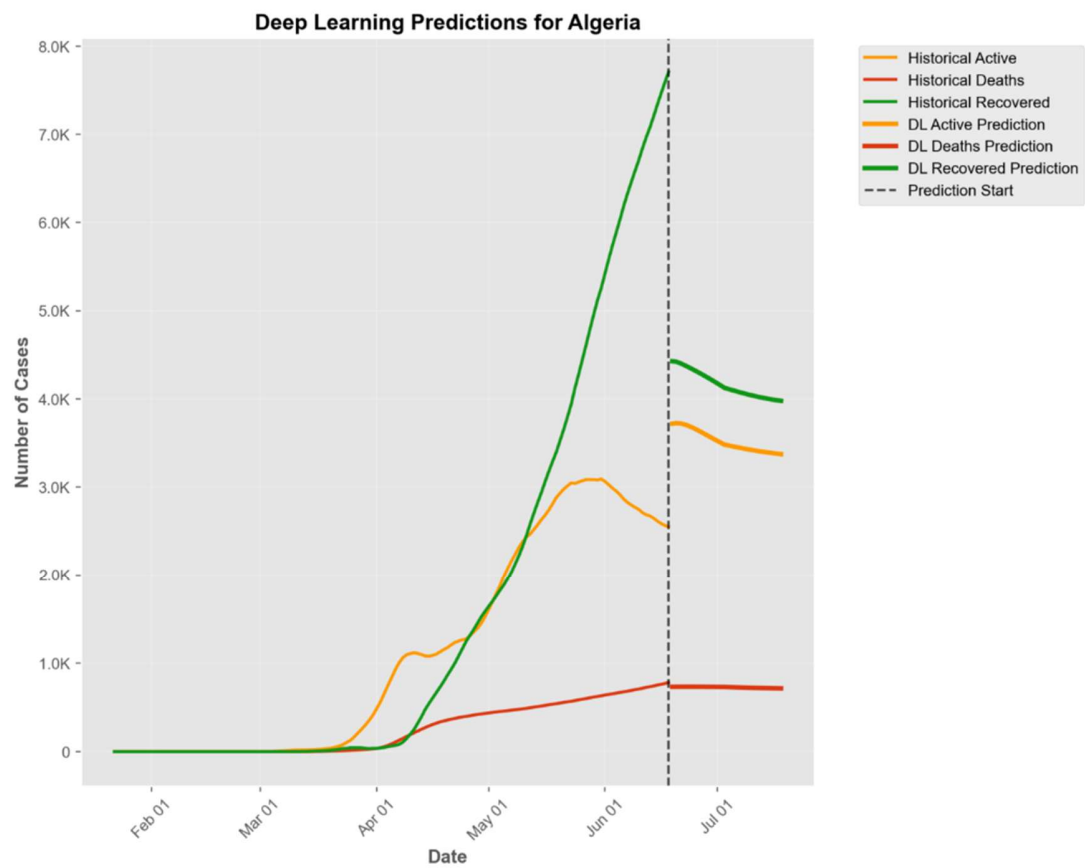


Figure 19: the deep learning model prediction

2.4. Comparison between the results of the models

The application, after generating and displaying the prediction results, has an entire section devoted to comparing the two models with each other. There are two ways in which this comparison format is presented:

- **Graphical comparison:** The application now draws comparative plots for each major category, namely active cases, deaths, and recoveries. These plots are colored side by side with predictions from the SIR model to the LSTM model so that users can easily see the differences and similarities in the forecasted trends (Figure 6, Figure 7, Figure 8)

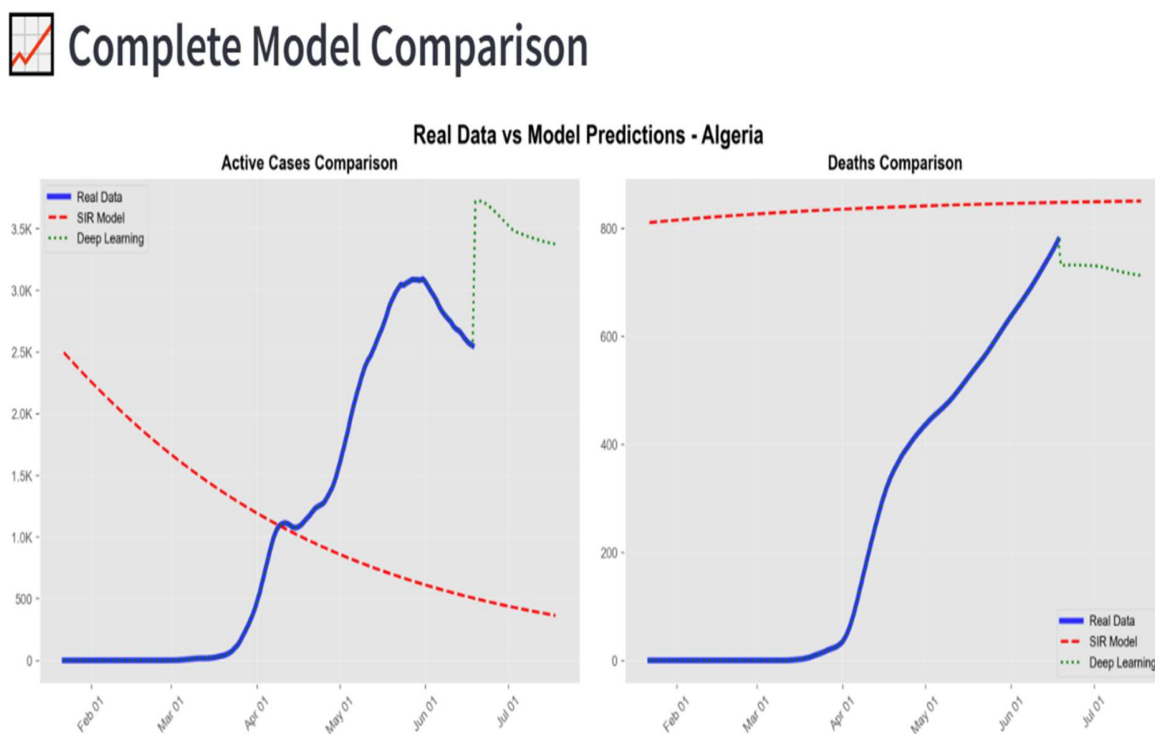


Figure 20: Graphical comparison Active cases & deaths

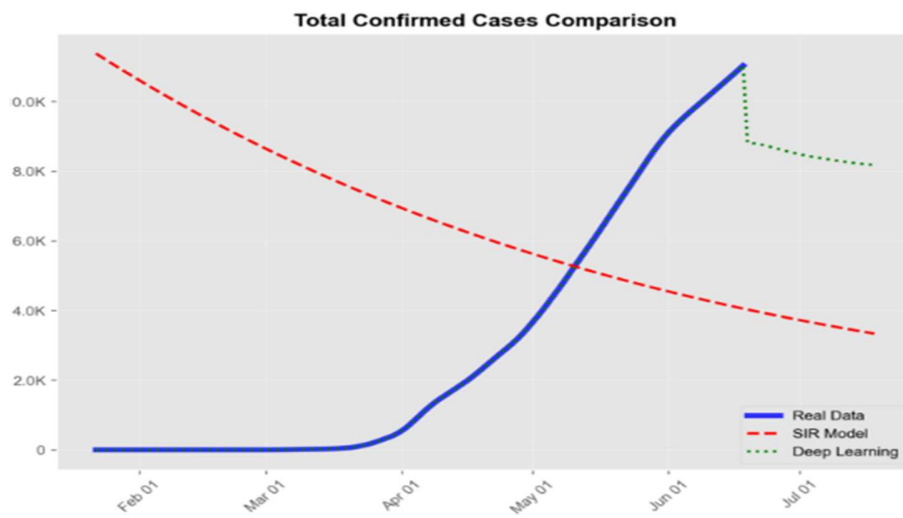


Figure 21: Graphical of confirmed cases comparison

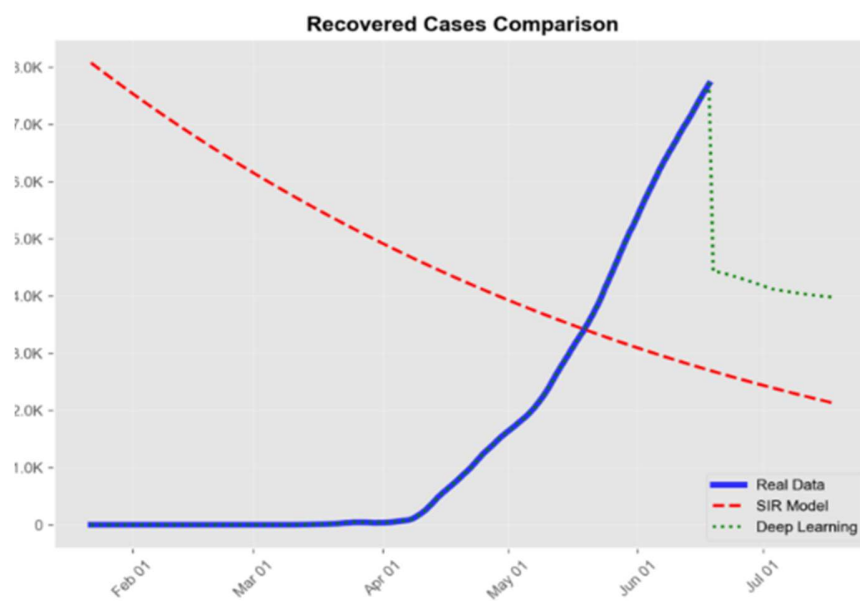


Figure 22: Graphical of Recovered cases comparison

- **Textual comparison:** If some users find it difficult to interpret graphical data, the application also provides them with a textual summary of the differences. This explanation looks at the key dissimilarities between the models' predictions in a concise manner, going over the implications depicted in the graphical format (Figure 9).

Prediction Summary Report

Current vs Predicted Comparison

Current Active Cases	Current Deaths	Current Recovered
2,496	811	8,078
Avg Predicted Active	Avg Predicted Deaths	Avg Predicted Recovered
1,974	759	3,801

Growth Rate Analysis

Active Growth Rate	Deaths Growth Rate	Recovery Growth Rate
-20.88%	-6.40%	-52.94%

Figure 23: Textual comparison

2.5. Interactive Report Generation

Finally, the application generates a comprehensive report summarizing the selected country's COVID- 19 data and the corresponding predictions. This report includes key forecasted values such as predicted active cases, predicted deaths, and predicted recoveries. It also provides an analysis indicating whether each of these values is expected to increase or decrease over time. The report presents the results for both the SIR model and the deep learning (LSTM) model, allowing for a side-by-side comparison. Additionally, the application offers the option to download the full report for offline viewing or further analysis (Figure 10).

COVID-19 Forecast Summary

Key Findings

- Current Active Cases: 19,326
- Current Deaths: 546
- Current Recovered: 7,660

Projections by July 18, 2020

- Predicted Active Cases: 13,147 (Growth: -31.97%)
- Predicted Deaths: 670 (Growth: 22.88%)
- Predicted Recovered: 4,567 (Growth: -40.37%)

Model Predictions Comparison

SIR Model:

- Active: 1,034
- Deaths: 800
- Recovered: 5,554
- R_0 : 1.00

Deep Learning Model:

- Active: 25,260
- Deaths: 541
- Recovered: 3,581

Model Agreement

- Active Cases Difference: 24,226
- Deaths Difference: 259
- Recovered Difference: 1,972

Figure 24: The final report

3. Technologies Used

All that was used in making the app was deep data science at its best stacked with web technologies in a way that ensures these components are robust while they apparently present a user-friendly interface.

3.1. Core Framework

Streamlit is a Python library used to build web applications with hardly any code. With this framework, UI components and widgets and state management are handled in the most elegant way possible. The framework also provides tools for interactive data visualization, which brings the analysis to much more alive form for users. Consequently, quick build and deployment of applications germinate, and the iterations or improvements can keep flowing.



Figure 25: Streamlit

3.2. Data Processing and Analysis

- **Pandas:** The application uses Pandas to manipulate and analyze the data in Python, in conjunction with the DataFrame operations which work well with time series. This library will allow cleaning and transforming the data into forms required for modeling with use cases that may also include aggregation and grouping operations to pull up inferred knowledge from unwieldy sets of data.



Figure 26: Pandas

- **NumPy:** Provides numerical array computing support, with fast array operations managing data processing. Its mathematical functionalities allow complicated calculations required for epidemiological simulation. Random number generation in the library supports the simulation to quantify prediction uncertainty.



Figure 27: NumPy

- **SciPy:** The Basis library contributes scientific computing capabilities, with its being outfitted with a differential equation solver for implementing the SIR model. It also has a nice differential evolution function for global optimization when fitting parameters. Further integration and optimization utilities contribute in favor of an application-oriented analysis.



Figure 28: SciPy

3.3. Deep learning

- **TensorFlow and Keras:** With the very application backbone in deep learning, LSTM networks would be implemented to capture temporal information in pandemic data. These frameworks school you in model training and evaluation through strong optimization methods. Their means of controlling regularization stop overfitting and promote good generalization of the model.



Figure 29: TensorFlow

- **Scikit-learn:** provides essential machine learning tools, such as data preprocessing features with MinMaxScaler to normalize input data. It allows for the implementation of meta-models by means of Ridge regression in ensemble approaches. Other features include performance metrics to evaluate the prediction accuracy and reliability of the models.



Figure 30: Scikit-learn

3.4 Visualization

- **Matplotlib:** serves as a core plotting library where all visualizations are first created by figure and axes management. Design and styling options exist for various types of plots to ensure clarity and informativeness. It supports the rendering of static visualizations, which serve as visual elements in the dashboard.



Figure 31: Matplotlib

- **Seaborn:** offers data visualization with an enhanced decorative touch that beautifies plot information. Color palette management by this library ensures the production of pretty and meaningful visuals. Its statistical visualization supports the transmission of complex relationships into comprehensible stories within the data.



Figure 32: Seaborn

4. Future Enhancements

Potential improvements for future application releases would include some main fields for development. Enhancements in modeling would mean considering more complex and suitable compartmental models like SEIR and SEIDR that allow for additional disease states. To improve the transmission modeling, mobility data would have to be considered, which allows us to include population movement patterns. The models should also be fed with vaccination data so that immunization campaigns' impacts could be considered. External factors could also be fed into the system for a better estimation, including weather and policy interventions. On the technical side, the focus could be directed towards cloud deployment to allow wider accessibility across the globe. A database as an integration would allow faster data access and more efficient storage. Development of API would enable programmatic access for integration into other systems. Automatic refreshing of data from authoritative sources would ensure that analysis is rendered with the most recent information available. User experience- wise, multi-country comparison features would help in gauging relative progression of the pandemic. Variable-based custom alert settings would monitor for when critical changes hit the metrics. User accounts would nurture configurations for frequent analysis. Optimization for a mobile user interface would foster multi-platform accessibility. Other analytical capabilities could locate regional hotspots identifying areas of concern. It could also analyze the effectiveness of interventions.

5. Conclusion

The COVID-19 Analysis and Prediction Dashboard showcase the strengths that can arise from harmonizing traditional epidemiological modeling with advanced machine learning algorithms. Essentially, through the implementation of an improved SIR model together with deep learning forecasts, the application makes a deeper, more accurate prediction that considers the fact that disease dynamics change with the passage of time. Using the Streamlit framework, developers have been able to rapidly build an interactive user interface that brings elaborated modeling to the layman. An array of visualization options and customizations exist for dissecting the historical data in great detail-as well as the forecast

This strategic application hence becomes an asset in comprehending the unfolding of pandemics and could find its way into the form of modifications for surveillances of other infectious diseases or any challenge that relates to public health.

General conclusion

The COVID-19 pandemic underscored the critical role of Artificial Intelligence (AI) in epidemic modeling, offering unprecedented capabilities in forecasting, containment, and policy optimization. AI-driven models have revolutionized traditional epidemiological approaches by integrating, real-time data streams, machine learning (ML), and deep learning (DL) to enhance accuracy and adaptability in predicting disease spread.

AI models such as PandemicLLM, developed by Johns Hopkins and Duke universities,

outperformed conventional forecasting methods by incorporating diverse data types demographics, genomic surveillance, public health policies, and epidemiological trends to predict infection surges and hospitalization rates weeks in advance .

Large language models (LLMs) and generative AI enabled dynamic reasoning, adapting to emerging variants and policy changes, which traditional statistical models struggled with

AI-powered tools SEIR models facilitated early outbreak detection by analyzing open source data, mobility patterns, and environmental factors (e.g., temperature, humidity).

AI assisted in evaluating the effectiveness of lockdowns, mask mandates, and vaccine distribution strategies. For instance, multi-linear regression models correlated COVID-19 spread with climatic conditions, aiding seasonal preparedness.

Predictive analytics helped hospitals allocate resources by forecasting ICU demand and mortality risks using clinical markers like CRP, LDH, and D-dimer levels.

Data Limitations: Early pandemic models suffered from biased or incomplete data, affecting reliability.

Privacy Concerns: AI's reliance on personal data (e.g., contact tracing apps) raised issues about surveillance and individual freedoms.

Interpretability: Many AI models operate as "black boxes," limiting transparency for policymakers.

Global AI Surveillance Networks: Collaborative platforms could enable real-time, cross-border epidemic monitoring.

Integration with Genomic Sequencing: AI can predict variant dominance, as seen in a University of Florida study that identified 11 COVID-19 variants 10 weeks before official CDC classification.

Ethical Frameworks: Balancing AI efficacy with privacy rights and equity remains a

priority for future pandemic responses.

Final Remarks:

AI has proven indispensable in modeling COVID-19, transforming reactive public health measures into proactive, data-driven strategies. While challenges persist, the fusion of AI with epidemiology promises a more resilient global health infrastructure, capable of mitigating future pandemics with greater speed and precision. Continued advancements in explainable AI, international data-sharing, and ethical governance will be pivotal in harnessing this potential.

References

- [1]. Ben Hassen, H., Elaoud, A., Ben Salah, N., & Masmoudi, A. (2020). A SIR-Poisson model for COVID-19: Evolution and transmission inference in the Maghreb central regions. *Arabian Journal of Science and Engineering*, 46, 93-102. <https://doi.org/10.1007/s13369-020-04792-0>
- [2]. Bentout, S., Chekroun, A., & Kuniya, T. (2020). Parameter estimation and prediction for

coronavirus disease outbreak 2019 (COVID-19) in Algeria. *AIMS Public Health*, 7(2), 306-318. <https://doi.org/10.3934/publichealth.2020026>

[3]. Brown, J. R., & Hersh, A. (2020). Investor behavior in the tech sector: A comprehensive study. *Harvard Business Review*, 98(5), 85-94.

[4]. Burden, R. L., Faires, J. D., & Burden, A. M. (2015). *Numerical analysis*. Cengage Learning.

[5]. Castillo-Chavez, C., Feng, Z., & Huang, W. (2002). On the computation of R_0 and its role in global stability. In *Mathematical approaches for emerging and reemerging infectious diseases: An introduction* (pp. 229–250). Springer.

[6]. Cheynet, E. (2020, April). Generalized SEIR epidemic model (fitting and computation). GitHub. <https://www.github.com/ECheyne/SEIR>

[7]. Chowdhury, R. H. (2024). Advancing fraud detection through deep learning: A comprehensive review. *World Journal of Advanced Engineering Technology and Sciences*, 12(2), 606-613.

[8]. Chowdhury, R. H. (2024). AI-driven business analytics for operational efficiency. *World Journal of Advanced Engineering Technology and Sciences*, 12(2), 535-543.

[9]. Deng, C., & Liu, Q. (2012). A computer virus spreading model with nonlinear infectivity on scale-free network. In *International Conference on Information Sciences, Machinery, Materials and Energy* (pp. 1684–1688). Atlantis Press.

[10]. Diekmann, O., Heesterbeek, J. A. P., & Metz, J. A. J. (1990). On the definition and the computation of the basic reproduction ratio R_0 in models for infectious diseases in heterogeneous populations. *Journal of Mathematical Biology*, 28(4), 365-382.

[11]. Djidjou-Demasse, R., Selinger, C., & Sofonea, M. T. (2020). Épidémiologie mathématique et modélisation de la pandémie de COVID-19: Enjeux et diversité. *Revue Francophone de Laboratoire*, 2020(526), 63–69.

[12]. Ferguson, N. M., Cummings, D. A. T., Fraser, C., Cajka, J. C., Cooley, P. C., & Burke, D.

- S. (2006). Strategies for mitigating an influenza pandemic. *Nature*, 442(7101), 448-452.
- [13]. Ferretti, L., Wymant, C., Kendall, M., Zhao, L., Nurtay, A., & Abeler-Dörner, L. et al. (2020). Quantifying SARS-CoV-2 transmission suggests epidemic control with digital contact tracing. *Science*, 368(6491), eabb6936.
- [14]. Funk, S., & Jansen, V. (2012). The talk of the town: Modelling the spread of information and changes in behaviour. In *Modeling the interplay between human behavior and the spread of infectious diseases* (pp. 93–102). Springer.
- [15]. Gan, C., Yang, X., Liu, W., Zhu, Q., & Zhang, X. (2012). Propagation of computer virus under human intervention: A dynamical model. *Discrete Dynamics in Nature and Society*. Hindawi Publishing Corporation, ID106950.
- [16]. Gulia, N., & Gulia, L. (2021). AI and epidemiology: A critical review of the literature. *Journal of Epidemiology and Global Health*.
- [17]. Gostin, L. O., & Friedman, E. A. (2020). The global health crisis: Addressing the COVID-19 pandemic and strengthening global health security. *JAMA*.
- [18]. Heffernan, J. R., Smith, R. J., & Wahl, L. M. (2005). Perspective on the basic reproductive ratio. *Journal of the Royal Society Interface*, 2, 281–293.
- [19]. Hethcote, H. W. (1994). A thousand and one epidemic models. *Frontiers in Theoretical Biology*, 100, 504–515.
- [20]. Hethcote, H. W. (2000). The mathematics of infectious diseases. *SIAM Review*, 42(4), 599–653.
- [21]. Hethcote, H. W. (2008). The basic epidemiology models: Models, expressions for R_0 , parameter estimation, and applications. In *Mathematical understanding of infectious disease dynamics* (pp. 1–61).
- [22]. Kamgang, J. C., & Sallet, G. (2008). Computation of threshold conditions for

epidemiological models and global stability of the disease-free equilibrium (DFE). *Mathematical Biosciences*, 213, 1–12.

[23]. Kermack, W. O., & McKendrick, A. G. (1927). A contribution to the mathematical theory of epidemics. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 115(772), 700–721.

[24]. Liu, Y., et al. (2020). Artificial intelligence in COVID-19 research: A review of AI applications in the detection, treatment, and monitoring of the virus. *Journal of Medical Internet Research*, 22(10), e20856.

[25]. Nguyen, T. M., et al. (2020). Artificial intelligence in epidemiology: Current trends and future challenges. *International Journal of Environmental Research and Public Health*.

[26]. Shahid, M. A., & Haider, Z. (2020). Application of deep learning in epidemiological research. *International Journal of Environmental Research and Public Health*.

[27]. Van den Driessche, P., & Watmough, J. (2002). Reproduction numbers and sub-threshold endemic equilibria for compartmental models of disease transmission. *Mathematical Biosciences*, 180, 29–48.

[28]. Howard Howie Weiss. The sir model and the foundations of public health. *Materials mathematics*, pages 0001–17, 2013.

[29]. BR Laxmi Sree and MS Vijaya. A weighted mean square error technique to train deep belief networks for imbalanced data. 2018.

[30]. Alessandro Comunian, Romina Gaburro, and Mauro Giudici. Inversion of a sir-based model: A critical analysis about the application to covid-19 epidemic. *Physica D: Nonlinear Phenomena*, 413:132674, 2020.

[31]. Ian Cooper, Argha Mondal, and Chris G Antonopoulos. A sir model assumption for the spread of covid-19 in different communities. *Chaos, Solitons & Fractals*, 139:110057, 2020.

[32]. Luyu Zhou, Chun Zhao, Ning Liu, Xingduo Yao, and Zewei Cheng. Improved lstm-based deep learning model for covid-19 prediction using optimized approach. *Engineering*

applications of artificial intelligence, 122:106157, 2023.

[33]. Zhiheng Huang, Wei Xu, and Kai Yu. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*, 2015.

[34]. Usharani Bhimavarapu, Gopi Battineni, and Nalini Chintalapudi. Improved optimization algorithm in lstm to predict crop yield. *Computers*, 12(1):10, 2023.

[35]. Burak Güllmez. Stock price prediction with optimized deep lstm network with artificial rabbits optimization algorithm. *Expert Systems with Applications*, 227:120346, 2023