Faculty of Science
Department of Mathematics
and Computer Science

Port Said University

Forecasting in Various Applications Using Hidden Markov Models

By:

Mosaad Mohammed Mohammed El Gharib Hendam

B.Sc. in Mathematics in 2010 and Diploma in Statistics in 2021.

Thesis

Submitted for the partial fulfillment of the requirements of a master's degree of Science in Mathematical Statistics

Supervisors

Prof. Mohammed Mohammed El Genidy

Professor of Mathematical Statistics,
Department of Mathematics and Computer Science,
Faculty of Science,
Port Said University, Egypt

Dr. Khaled Mahfouz Abd-El Wahab

Lecturer of Mathematical Statistics,
Department of Mathematics and Computer Science,
Faculty of Science,
Port Said University, Egypt

2024

Faculty of Science
Department of Mathematics
and Computer Science

Port Said University

Title:

Forecasting in Various Applications Using Hidden Markov Models

Name:

Mosaad Mohmmed Mohammed El Gharib Hendam

Date of Discussion: 24 / 9 /2024

SUPERVISORS

No.	Name	Position	Signature
1	**Prof. Mohammed Mohammed El Genidy**	Professor of Mathematical Statistics, Department of Mathematics and Computer Science, Faculty of Science,	
2	**Dr. Khaled Mahfouz Abd-El Wahab**	Lecturer of Mathematical Statistics, Department of Mathematics and Computer Science,Faculty of Science,	

Head of Mathematics and Computer Science Department

Prof. Mohammed Mohammed El Genidy

Vice Dean for Research and Post-Graduate Affairs

Dean of Faculty of Science

Prof. Mohammed Kamel Farh

ACKNOWLEDGMENTS

I thank Allah, to him be the glory forever, who gave me the strength and settled everything to accomplish this thesis. I would like to express my gratitude to all the members of the Department of Mathematics and Computer Science who have taught me about this wonderful field of science and inspired me to teach it. I am especially thankful to my Professor, Mohammed El Genidy, for his patient guidance, support, effort, and suggestions. I would also like to thank Dr. Khaled Mahfouz. for his instructions. I am grateful to all my friends at the Faculty of Science for their encouragement and unconditional support.

Contents:

List of Tables

List of Figure

Page

List of Abbreviations

HMMs	Hidden Markov Models
NWP	Numerical Weather Prediction
NOAA	The National Oceanic and Atmospheric Administration
ECMWF	European Centre for Medium-Range Weather Forecasts
WMO	World Meteorological Organization
GISS	Goddard Institute for Space Studies
JMA	Japan Meteorological Agency
WHO	The World Health Organization
CDC	Centers for Disease Control and Prevention
ECDC	European Centre for Disease Prevention and Control
CSSE	Center for Systems Science and Engineering
MCMC	Markov Chain Monte Carlo
PMFs	probability mass functions

List of Publications

Research Article

Hendam, M.M., Mahfouz, K. and Genedi, M., 2024. Prediction Using Markov Model and Hidden Markov Model for the States of Banknotes. Alfarama Journal of Basic & Applied Sciences, 5(2), pp.262-280.

https://doi.org/10.21608/AJB AS.2024.241492.1190

International Conference

Detecting counterfeit banknotes and their values using the Markov model and hidden Markov model

Mohammed El Genidy, Mosaad Hendam*, Khaled Mahfouz Department of Mathematics and Computer Science, Faculty of Science Port Said University, Port Said, Egypt.

For participation by oral presentation in the first international conference on green science hold in Port said, 24-25 September 2023.

Summary

This thesis aims to apply the Markov Model and the Hidden Markov Model (HMM) for predicting and verifying the values of currency notes, a subject of significant interest in the field of currency recognition. The challenge of authenticating and determining the value of currency notes is a major concern for researchers, particularly with the growing demand for accurate and reliable systems in digital financial platforms.

The Markov Model is a powerful tool for assessing the probabilities associated with currency values, relying on a sequence of transitional probabilities between different states, such as whether a note is genuine, damaged, or counterfeit. This model is useful for understanding and analyzing potential patterns that may emerge throughout the lifecycle of currency notes, such as frequency of use or exposure to damage.

However, in more complex scenarios, the true state of the currency may not be directly observable, necessitating the use of the Hidden Markov Model (HMM). This model enables the estimation of hidden states of currency based on observable data, such as the number of deposits or repeated usage in financial systems. Empirical studies have demonstrated the effectiveness of the HMM in providing accurate predictions about the future state of currency after a limited number of uses, thereby enhancing the precision of detection systems and reducing potential errors.

Moreover, the results of this research can be generalized to other fields that require forecasting future probabilities based on historical data, such as weather prediction and epidemic spread analysis. In the realm of weather forecasting, transition matrices derived from the Markov Model have shown their ability to predict future weather conditions by observing past patterns, such as rising temperatures after rainfall. However, the model faces

limitations, including its inability to account for seasonal or geographical variations. To address these limitations, the HMM comes into play, utilizing indirect indicators like the frequency of umbrella usage to provide more accurate weather forecasts.

In a completely different context, the use of both the Markov Model and HMM has contributed to analyzing the dynamics of COVID-19 spread. The HMM has proven particularly useful in demonstrating the significant role of vaccination in altering the trajectory of the pandemic and reducing the rate of virus transmission. By studying recorded cases and vaccination rates, the model has provided precise insights into how daily activities, such as vaccination rates, impact the spread of the virus, aiding in the improvement of public health strategies.

In conclusion, this thesis makes a substantial scientific contribution by applying the Markov Model and Hidden Markov Model to a range of predictive tasks, including currency evaluation, weather forecasting, and epidemic analysis. These findings support data-driven decision-making processes across various domains, from financial systems to public health and environmental management.

Chapter 1

Introduction

Chapter (1) Introduction

1.1 Overview

A significant concern in statistical modelling is establishing an accurate, estimated model as the foundation for inference. Markov Models and Hidden Markov Models (HMMs) stand out as essential statistical tools for predictive modelling across various disciplines, including speech recognition, bioinformatics, public health and meteorology [1]. The foundation of a Markov Model lies in its adherence to the Markov property, which states that the future state of a process depends only on its present state, not on the sequence of events that came before it [2]. This principle simplifies the prediction of future states in a sequence. Conversely, Hidden Markov Models enhance this approach by focusing on processes where the states are not directly observable. Instead, what is observable is the output influenced by these hidden states. HMMs are invaluable for revealing hidden processes that are not immediately apparent, like decoding the sequence of states in a speech signal or predicting weather based on indirect observations [3].

The determination of banknote values through the application of Markov models and HMM represents a sophisticated approach to currency valuation and authenticity assessment. By leveraging Markov models, probabilities associated with the values of banknotes can be effectively evaluated, providing insights into their monetary worth. Additionally, the utilization of HMM allows for the assessment of the probabilities of various states of banknotes, particularly after undergoing a series of transactions or being deposited multiple times. Through these models, significant numerical findings can be derived, contributing to a more comprehensive understanding of currency dynamics and facilitating informed decision-making processes in financial contexts [4].

Weather forecasting, a critical application of predictive modelling, utilizes a diverse array of models, each with unique strengths and methodologies that markedly differ from the Markov Model approach. Numerical Weather Prediction (NWP) models dominate this field, employing complex equations to simulate atmospheric conditions and offering detailed forecasts over short to medium ranges [5]. Statistical models, leveraging historical data, cater to long-term forecasting through regression analysis and other similar techniques. Ensemble models, integrating multiple forecasts, aim to mitigate uncertainties by presenting a spectrum of potential outcomes [6]. In contrast, the relatively simpler Markov Models focus on predicting weather through state transitions, optimizing for straightforward, short-term forecasts like rain probabilities. For a more comprehensive approach to weather forecasting, particularly for complex systems or long-term predictions, the advanced methodologies of NWP and Machine Learning are generally favored [7].

The endeavor to forecast the spread of COVID-19 underscores a pivotal integration of epidemiological insight and statistical modelling, with HMM playing a critical role. This segment delves into the application of HMMs in tracing and predicting the pandemic's trajectory, a theme ripe for exploration within academic theses or research papers [8]. The COVID-19 pandemic has highlighted the imperative for precise, adaptable forecasting models to anticipate the spread of infectious diseases accurately. The utility of HMMs in forecasting the trajectory of COVID-19 pivots on its methodological strengths, the challenges encountered, and its consequent implications for public health strategy [9].

HMMs are especially fitting for the task of modelling infectious diseases, thanks to their capacity to articulate the sequence of unobservable state transitions encompassing various stages of infection and their evolution over time [10]. In the context of COVID-19, these states could encompass a range from susceptible to exposed, to asymptomatic or symptomatic infections, and finally to recovery or death. The model harnesses observed data points like daily confirmed cases, hospitalizations, and fatality counts to infer the likelihood of transitions between these states. This modelling approach is instrumental in providing a nuanced understanding of the pandemic's spread, serving as a cornerstone for strategic planning and response in public health initiatives [11].

1.2 Thesis Objective

This thesis aims to use Markov models and HMM to determine banknote values and verify their authenticity, crucial for currency valuation. By assessing probabilities associated with banknote values and states, it seeks to deepen understanding of currency dynamics and facilitate decision-making in finance. Additionally, it aims to extend findings to enhance predictive capabilities in weather forecasting and other domains. Researchers seek statistical methods to achieve effective forecasting in areas as diverse as weather patterns and public health crises like COVID-19, leveraging advanced methodologies to navigate the complexities of these phenomena. Among the statistical tools at their disposal, HMM stands out for its robust capacity to model and predict dynamic systems with a high degree of accuracy [12]. In weather forecasting, HMMs are applied to develop a nuanced approach that transforms atmospheric conditions into a series of hidden states based on variables such as temperature, humidity, and pressure. This model, enriched with observable meteorological data, aims to significantly refine the precision and reliability of short-term forecasts. The underlying goal is to exploit temporal correlations within weather patterns, delivering essential predictions that support decision-making in agriculture, transportation, and disaster management. Efforts are also dedicated to optimizing model parameters, enhancing computational efficiency, and ensuring the seamless integration of these forecasts into existing meteorological frameworks to enable real-time strategic planning [13].

Similarly, in the public health domain, particularly in tackling the COVID-19 pandemic, HMMs provide a framework for accurately modeling the virus's transmission dynamics. By identifying hidden states that represent different stages of infection and combining this with critical epidemiological data such as case counts, testing rates, and population mobility patterns researchers aim to improve the accuracy and timeliness of their predictions. This enables public health officials, policymakers, and healthcare systems to make informed decisions about resource allocation, intervention planning, and risk assessment [14].

Furthermore, this strategy includes exploring methodologies to handle uncertainty and variability in the model parameters, assessing the impact of various public health interventions, and integrating HMM-based forecasts with existing epidemiological models [15].

1.3 Weather dataset.

Weather prediction relies heavily on accurate and comprehensive datasets, and there are several key sources where such data can be obtained. The National Oceanic and Atmospheric Administration (NOAA) is a primary source, offering a vast array of climate and weather data, including satellite imagery. Similarly, the European Centre for Medium-Range Weather Forecasts (ECMWF) provides highly accurate datasets used globally for weather forecasting [16]. The World Meteorological Organization (WMO) facilitates access to international meteorological data, while NASA's Goddard Institute for Space Studies (GISS) offers crucial datasets for climate analysis.

The UK Met Office and the Japan Meteorological Agency (JMA) also provide extensive weather data, with JMA being a notable source for Asian weather patterns. Beyond this, public data repositories like Kaggle and Google Dataset (www.kaggle.com) Search are valuable for machine learning projects, offering diverse weather datasets. For more targeted academic research, many universities and research centers maintain specialized meteorological data [17].

Additionally, commercial providers like Accu Weather and The Weather Company offer real-time weather data services, though often at a cost. Finally, open-source projects like Open Weather Map provide free access to weather data, catering to a broad range of users from developers to hobbyists. When selecting a dataset, it's important to consider geographical coverage, period, data frequency, and the specific meteorological variables required, always respecting the terms of use and data usage policies [18]. The dataset contains weather observations for Seattle City. As shown in Table 1.1

Table 1.1 The dataset contains weather observations for Seattle City.

Date	precipitation	Max Temp	Min Temp	Wind	Weather
1-1-2012	0.0	12.8	5.0	4.7	drizzle
2-1-2012	10.9	10.6	2.8	4.5	rain
3-1-2012	0.8	11.7	7.2	2.3	rain
⋮	⋮	⋮	⋮	⋮	⋮
30-12-2015	0	5.6	-1.0	3.4	sun
31-12-2015	0	5.6	-2.1	3.5	sun

Each column provides specific data points related to weather observations in Seattle City. The *Date* column records the date when the observation was made, allowing for chronological organization of the data. The *Precipitation* column quantifies the amount of rainfall recorded, measured in millimetres.

This information is crucial for understanding the frequency and intensity of precipitation events in the region.

The *Max Temp* and *Min Temp* columns provide insight into temperature variations within Seattle City. The *Max Temp* column indicates the highest temperature recorded during the observation period, while the *Min Temp* column denotes the lowest temperature. These temperature measurements are essential for analyzing daily temperature fluctuations and identifying trends over time.

The *Wind* column offers data on wind speed, measured in kilometers per hour. Wind speed plays a significant role in weather patterns and can impact various aspects of daily life, such as transportation and outdoor activities. Understanding wind patterns is essential for assessing weather conditions and their potential effects on the environment and human activities.

Finally, the *Weather* column describes the prevailing weather conditions at the time of observation using descriptive terms such as "drizzle" or "rain." This qualitative data provides additional context to complement the quantitative measurements recorded in other columns. By combining both quantitative and qualitative information, the dataset facilitates comprehensive analysis and interpretation of weather patterns in Seattle City.

Table 1.2 The 5-number summary for the numerical columns in the dataset.

Statistic	precipitation	Max Tem	Min Tem	Wind
Minimum	0.0	-1.6	-7.1	0.4
25th percentile	0.0	10.6	4.4	2.2
Median	0.0	15.6	8.3	3.0
75th percentile	2.8	22.2	12.2	4.0
Maximum	55.9	35.6	18.3	9.5

Table 1.2 shows the 5-number summary is a concise statistical summary that helps to understand the distribution of numerical data within a dataset. For the Seattle weather data, the 5-number summary reveals various aspects of the weather conditions. The precipitation has a median of 0.0 mm, indicating that at least half of the days recorded no precipitation, with the maximum precipitation recorded as 55.9 mm in a day, pointing to occasional heavy rainfall. Temperature data, both maximum (Max Temp) and minimum (Min Temp), shows a range from below freezing to quite warm, with values ranging from -1.6°C to 35.6°C for maximum temperatures and from -7.1°C to 18.3°C for minimum temperatures, reflecting the varied climate through different seasons. Wind speeds vary from a gentle 0.4 km/h to a brisk 9.5 km/h, with the median wind speed at 3.0 km/h, indicating generally mild conditions typical of the Seattle area. This summary provides a quick snapshot of the weather extremes and typical conditions, useful for understanding the climate variability in Seattle.

Fig 1.1 shows the line chart above illustrates the total monthly precipitation in Seattle City, capturing several years of data to reveal seasonal trends and variability in rainfall. Each point on the line represents the sum of precipitation for a specific month, allowing us to observe how rainfall accumulates over time within each year and across years. The visualization shows clear cyclic patterns, indicating the seasonal nature of precipitation in Seattle. Months with peaks suggest periods of high rainfall, which are critical for understanding the wettest times of the year, while the valleys denote drier months, providing a visual representation of the typical climatic rhythm in the region.

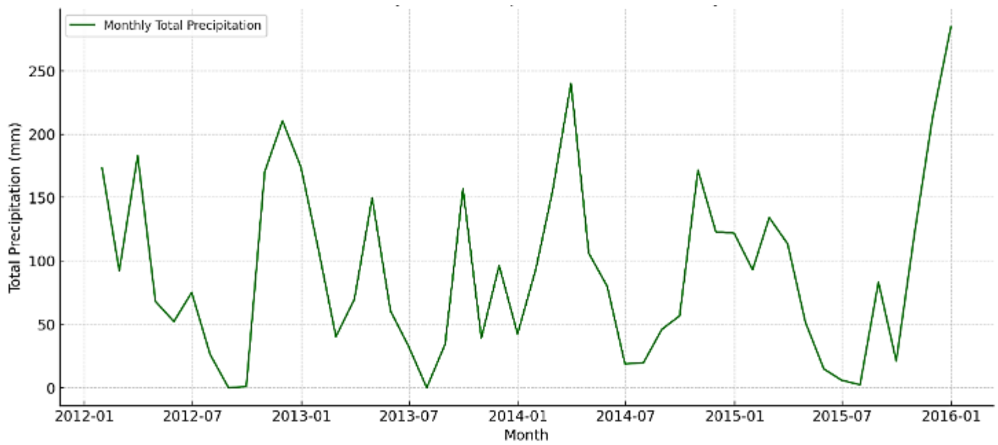

Figure 1.1 The total monthly precipitation in Seattle City.

Analyzing the chart, it's clear that certain periods consistently experience higher precipitation, which could align with Seattle's known rainy seasons in late fall and winter. Conversely, the lower points during other months highlight the typical dry spells, particularly evident in the summer months. This pattern of variation is essential for various stakeholders, including city planners, farmers, and residents, who rely on such information for planning and resource management. The line chart effectively encapsulates these trends, offering a straightforward yet powerful tool to gauge weather-related expectations and prepare for the impact of seasonal changes on day-to-day and long-term activities.

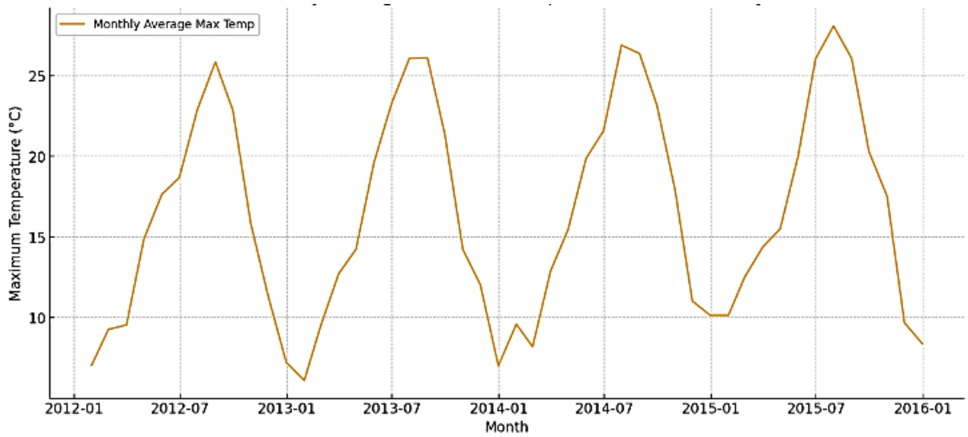

Figure 1.2 The monthly average maximum temperatures in Seattle City.

Fig 1.2 shows the line chart displayed above represents the monthly average maximum temperatures in Seattle City, reflecting the typical thermal patterns over several years. This visualization uses the average daily maximum temperatures recorded each month, depicted by the orange line, to demonstrate the warming and cooling trends throughout the year. Notably, the chart exhibits a sinusoidal pattern, peaking during the summer months when temperatures are highest and dipping during the winter months, consistent with the temperate maritime climate of Seattle.

From this chart, we can observe that the warmest months are generally in the middle of the year (June through August), where the maximum temperatures are noticeably higher. Conversely, the coldest months appear towards the beginning and end of the year (December through February), highlighting the seasonal temperature extremes. Such visualizations are vital for understanding climate behavior, aiding in everything from personal planning and clothing choices to energy management and agricultural decisions in the region. This line chart not only simplifies the comprehension of annual temperature cycles but also underscores the variability and predictability of weather patterns, essential for both residents and businesses in Seattle. Temperatures in Seattle. It highlights the general trend of temperatures throughout the year, with all data averaged across the available years. This line chart, plotted with the monthly averages on the y-axis and the months of the year on the x-axis, shows a typical sinusoidal pattern which is characteristic of temperate climates. Temperatures reach their lowest points during the winter months, particularly in January and December, and peak during the summer, with July typically being the warmest month. Fig 1.3 shows visualization in particularly useful for understanding the climatic pattern of minimum temperatures, which can aid in planning for agricultural activities, energy consumption forecasting, and even in preparing public health advisories for extreme weather. Fig 1.4 shows the line chart showing the overall average monthly wind speeds in Seattle. The trend line illustrates how wind speeds vary from month to month across all the years included in the dataset.

The line chart, marked by points for each month and connected by a line, reveals the variations in wind speed, with higher speeds generally observed in the colder months. The peaks in wind speed during the fall and early spring can be associated with transitional weather patterns, where shifts between high and low-pressure systems are more frequent and intense. This chart is useful for sectors that are sensitive to wind conditions, such as maritime operations, aviation, and even event planning, providing essential insights into expected wind conditions throughout the year.

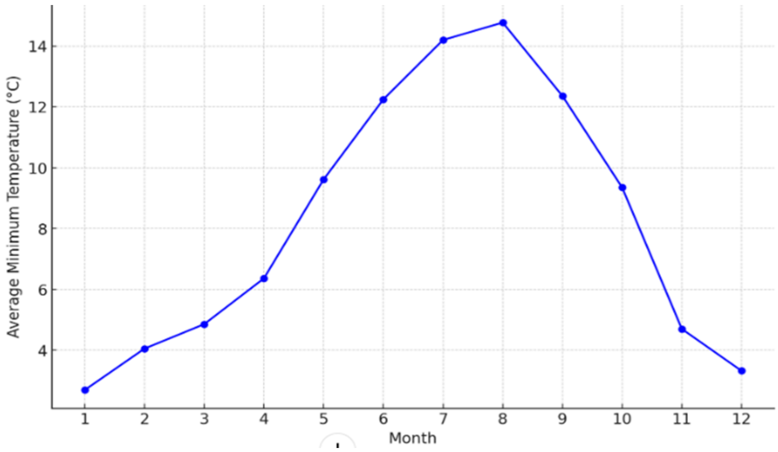

Figure 1.3 The average monthly minimum temperatures in Seattle

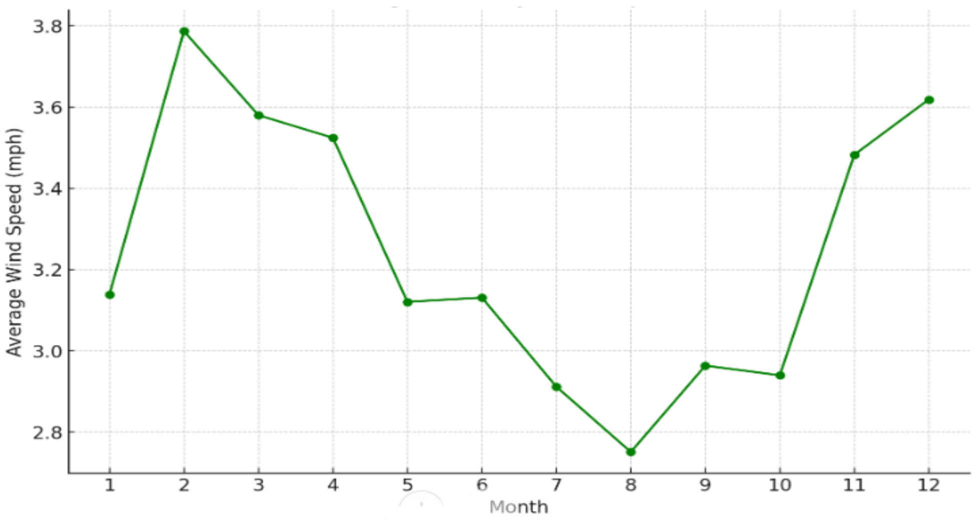

Figure 1.4 The average monthly wind speeds in Seattle.

1.4 COVID-19 dataset.

The COVID-19 pandemic has necessitated the compilation of extensive datasets to predict and analyze the spread of the virus. Key sources for COVID-19 prediction datasets include government health departments and international health organizations. The World Health Organization (WHO) provides global data on COVID-19 cases, deaths, and vaccination rates, essential for worldwide trend analysis [19].

Similarly, the Centers for Disease Control and Prevention (CDC) in the United States and the European Centre for Disease Prevention and Control (ECDC) offer detailed datasets specific to their regions, including infection rates, demographic information, and the impact of public health interventions. The COVID-19 Data Repository by the Center for Systems Science and Engineering (CSSE) at Johns Hopkins University has become a widely referenced source, offering daily updated global data on the pandemic [20].

Academic and research institutions around the world also contribute by sharing data from their independent studies, often focusing on specific aspects such as transmission dynamics, the efficacy of vaccines, or the impact of variants [21].

Furthermore, platforms like Kaggle and Google Dataset Search provide access to a variety of COVID-19 datasets, which are particularly useful for data-driven research and machine-learning projects. These datasets are invaluable for modelling the pandemic's trajectory, understanding its impact, and informing policy decisions. However, it's crucial to consider the timeliness, accuracy, and completeness of the data. As well as respecting privacy and ethical guidelines when using these datasets for analysis or predictive modelling. https://www.kaggle.com/datasets/jhossain/covid19-in-bangladesh. The data file provides detailed information on COVID-19 in Bangladesh, focusing on daily reports. It includes several key columns: Date, indicating the record date; Confirmed, indicating the number of confirmed COVID-19 cases reported on that date; Deaths, representing the reported deaths due to COVID-19 on the same date; and recovered, indicating the number of patients who recovered from COVID-19 on that particular date.

This dataset enables tracking and analysis of the pandemic's progression and the impact of public health measures over time in Bangladesh.

Table 1.3 The first and last 5 rows of the COVID-19 data for Bangladesh.

Date	Confirmed	Deaths	Recovered
8-3-2020	3	0	0
9-3-2020	3	0	0
⋮	⋮	⋮	⋮
12-4-2020	621	34	39
13-4-2020	803	39	42

Table 1.3 shows the first and last 5 rows of the COVID-19 data for Bangladesh, spanning from March 8, 2020, to April 13, 2020. The Date column indicates the recording dates, capturing both early and later phases of the pandemic. Initially, there were few confirmed cases, with only 3 reported daily in the first five entries. However, by early April, the numbers surged significantly, reflecting rapid viral transmission. The Deaths column initially showed no fatalities but rose to 39 by April 13, 2020, illustrating the increasing severity of the outbreak. Meanwhile, the Recovered column began with zero recoveries but gradually increased as more patients recuperated from the virus. Focusing attention on the dataset's initial and recent snapshots, highlighting the pandemic's progression over time in Bangladesh.

This table is useful for observing the trends of COVID-19 in terms of case counts, recoveries, and fatalities over time, especially to understand how quickly the virus spread and its impact over a month.

Table 1.4 The 5-number summary of the confirmed cases, deaths, and recoveries in Bangladesh.

Statistic	Confirmed	Deaths	Recovered
Min	30	0	0
25%	10	0	0
50%	44	5	11
75%	70	8	30
Max	803	39	42

The 5-number summary provides a comprehensive overview of COVID-19 data in Bangladesh, revealing the breadth and impact of reported cases, deaths, and recoveries throughout the observed period. The Minimum, which marks the lowest count of confirmed cases at 3, underscores the early stages of the pandemic when initial infections were just emerging. The First Quartile (25%) illustrates days with minimal impact, where 25% reported 10 or fewer confirmed cases, reflecting early containment or limited spread efforts. The Median (50%) indicates a typical day during the pandemic with 44 or fewer confirmed cases, reflecting moderate community spread and fatalities of 5 or fewer deaths and 11 or fewer recoveries. The Third Quartile (75%) shows a significant impact of the virus on most days with 70 or fewer confirmed cases, with deaths and recoveries each ranging up to 8 and 30 respectively.

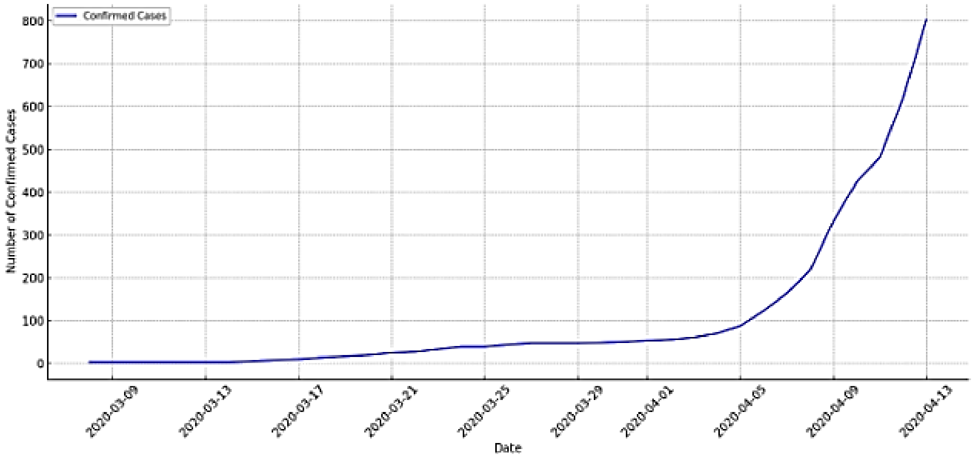

Figure 1.5 The confirmed COVID-19 cases over time in Bangladesh.

Figure 1.5 offers a visual representation of the progression of confirmed COVID-19 cases in Bangladesh from March 8, 2020, to the latest available date in the dataset. The x-axis denotes the timeline with specific dates, providing temporal context, while the y-axis quantifies the cumulative count of confirmed cases, ranging from the lowest to the highest observed counts. Initially showing a gradual increase, suggesting a slow rise in case numbers early on. As time advances, the line steepens, indicating a significant surge in confirmed cases, possibly due to increased viral spread, heightened testing, or changes in reporting protocols.

Fluctuations or peaks in the line may signify localized outbreaks or policy shifts, such as the implementation or easing of public health measures. Each phase of the line offers insights into the pandemic's dynamics in Bangladesh, illustrating how the virus has spread and responded to various interventions. This visualization is crucial for policymakers, health officials, and the public, aiding in the assessment of intervention effectiveness and the formulation of future strategies to manage and mitigate the impact of COVID-19 effectively

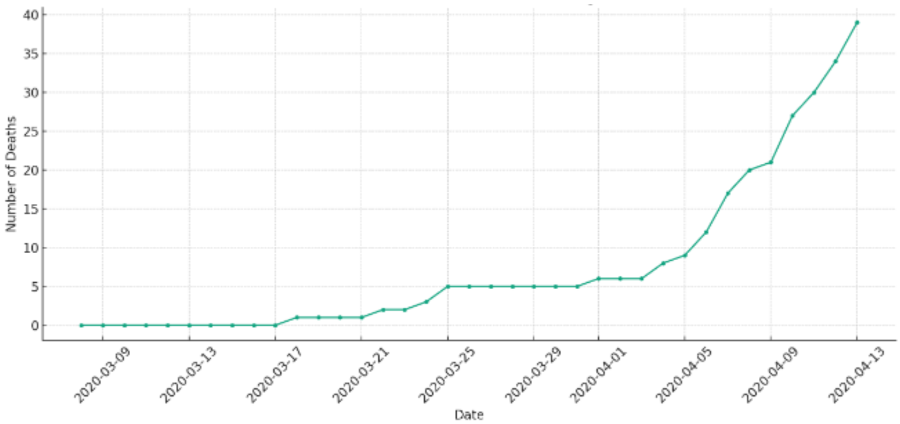

Figure1.6 The trend of COVID-19 deaths over time in Bangladesh.

Figure 1.6 illustrates the trajectory of COVID-19 deaths in Bangladesh since the onset of the pandemic in March 2020. The x-axis marks the timeline, spanning from March 8, 2020, onward, providing a chronological view of how the death toll has evolved. The y-axis depicts the cumulative number of daily deaths attributed to COVID-19. The line graph shows fluctuations in the death toll, with the slope indicating the rate of increase in deaths over time. Steeper slopes indicate periods of rapid escalation in fatalities, highlighting severe impacts of the virus. Conversely, flattening segments suggest stabilization or periods without new reported deaths, potentially reflecting effective control measures. Each data point on the line corresponds to the total deaths reported on a specific date, offering a detailed day-to-day perspective on the pandemic's mortality impact and pinpointing critical dates of significant changes. This visualization serves as a vital tool for public health officials, policymakers, and researchers to assess intervention effectiveness and plan future strategies to manage and mitigate the pandemic's health system strains effectively.

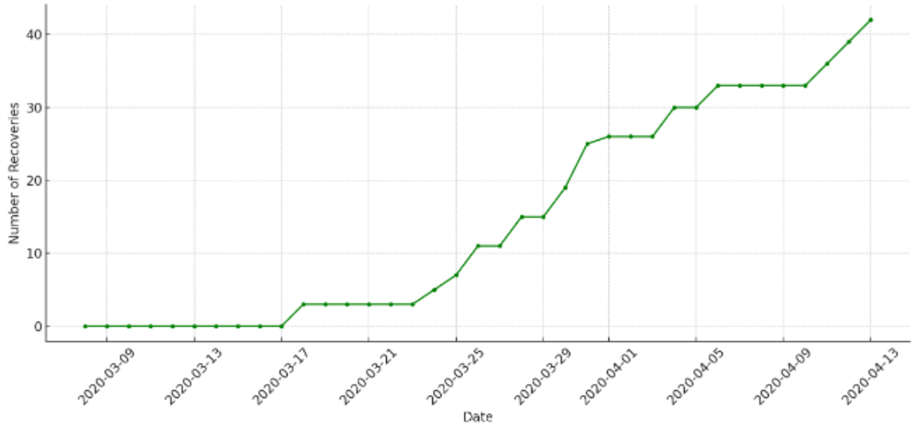

Figure 1.7 The trend of COVID-19 recoveries in Bangladesh.

Figure 1.7 displays the cumulative trend of COVID-19 recoveries in Bangladesh since the pandemic's onset in early March 2020. Using the x-axis to denote dates from the first recorded case, and the y-axis to quantify cumulative recoveries reported each day, the chart offers a longitudinal view of recovery progression. The green line tracks these cumulative recoveries, illustrating a consistent upward trajectory indicative of increasing recoveries over time. This reflects both the natural recovery process and the effectiveness of healthcare interventions. Variations in the line's slope highlight periods of accelerated or slower recovery rates, potentially influenced by treatment advancements or fluctuations in disease severity. Integrating recovery data with other metrics such as infection and mortality rates provides a holistic view of the pandemic's trajectory, aiding policymakers and healthcare professionals in refining strategies to manage and mitigate its impact effectively.

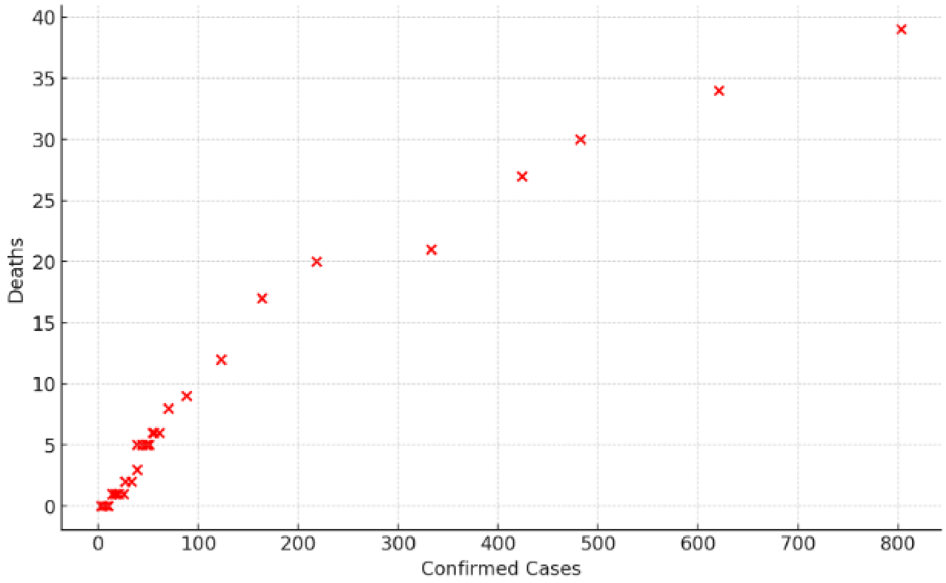

Figure 1.8 The relationship between the number of confirmed COVID-19 cases and the number of deaths in Bangladesh.

The scatter plot in Figure 1.8 depicting the relationship between confirmed COVID-19 cases and deaths in Bangladesh serves as a crucial tool for health officials and policymakers. It reveals a clear positive correlation, indicating that as the number of confirmed cases increases, so does the number of deaths. This correlation underscores the impact of case numbers on mortality rates, highlighting scenarios where healthcare systems may face strain or where effective treatments are not universally accessible. The plot shows a concentration of data points at lower levels, suggesting periods with relatively low case and death counts, but as case numbers rise, the data points scatter more widely. This dispersion may reflect varying death rates or disparities in healthcare access and management as the pandemic progresses. Furthermore, outliers in the plot, such as instances of high case counts with relatively few deaths, offer insights into effective interventions or demographic factors influencing disease outcomes. Analyzing these patterns can guide future interventions and preparedness efforts to effectively manage and mitigate the ongoing impact of COVID-19.

Overall, this scatter plot is a valuable tool for understanding the direct impact of the spread of the virus in terms of mortality. It can help health officials and policymakers evaluate the lethality of the virus at different stages of case prevalence and assess the effectiveness of medical interventions and public health measures. The scatter plot above depicts the relationship between confirmed COVID-19 cases and the number of recoveries in Bangladesh as in Fig 1.9. Observing such a relationship helps to understand how the recovery rates correspond with the growth in confirmed cases.

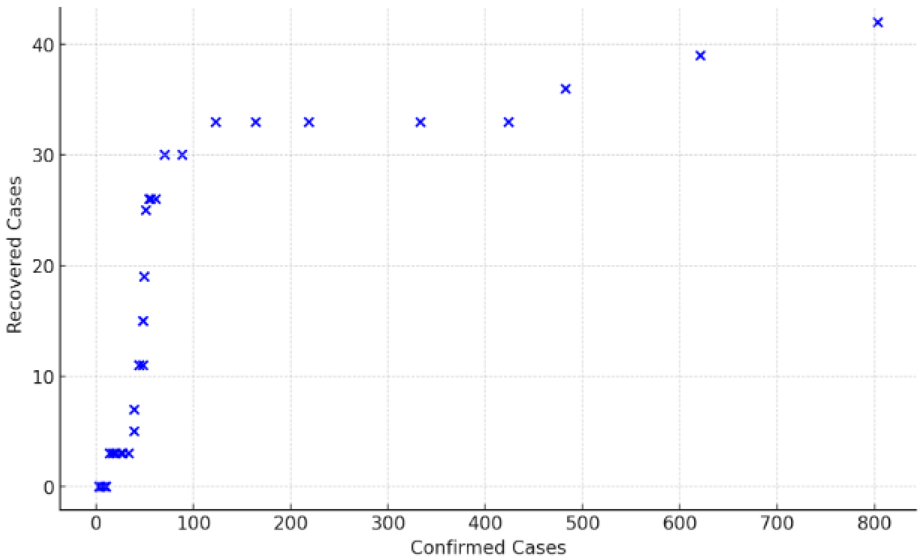

Figure 1.9 The relationship between confirmed COVID-19 cases and the number of recoveries in Bangladesh.

The relationship between the number of confirmed COVID-19 cases and the number of recoveries in Bangladesh shows that as more people get infected, more also get better. This makes sense during a pandemic, where higher cases eventually lead to more people recovering from the virus. The data points on the graph are mostly grouped together at lower levels, showing times when both the number of cases and recoveries were low. But as the number of confirmed cases goes up, the points spread out more. This suggests that different factors like how well treatments work, how many people can get healthcare, and how healthy people are affecting how many recover. Overall, the graph shows that the healthcare system has been helping people recover as the number of cases keeps growing. This is good news because it means the healthcare system is managing to treat people well even as more and more get sick.

1.5 Banknotes dataset.

To generate values between (0) and (1) they obtained using the Excel program by the code(= RAND () *(b-a)+a, a=0 and b=1).

Another way by using Mathematica by the code:

(Table [Random [Real, {a, b}], {c}]), where c is the number of required probabilities. Applying the previous code in Excel, then the following probabilities were obtained. Suppose the states of banknotes A and B.

$P(A|A)$ = 0.8, $P(B|A) =$ 1- $P(A|A)$= 0.2 $P(B|B)$ = 0.6

$P(A|B) =$ $P(B|B)$= 0.4 , $P(A) = \frac{2}{3}$, $P(B) = \frac{1}{3}$

$P(A|A)$: represents the probability of state A at time t+1, given that the system is currently in state A at time t.

$P(B|A)$: represents the probability of state B at time t+1, given that the system is currently in state A at time t.

Chapter 2

Fundamental Concepts

Chapter (2) Fundamental Concepts.

2.1 What is the Forecasting?

Forecasting is the process of making predictions or estimates about future events or conditions based on past and present data. It involves analyzing historical data, identifying trends or patterns, and extrapolating these findings to predict future outcomes. Forecasting is crucial across various domains and disciplines to aid decision-making and planning [22].

2.2 The difference between forecasting and prediction.

Forecasting and prediction are often used interchangeably, but they have distinct meanings. Forecasting involves using historical data, statistical methods, and models to estimate future events, often considering trends and patterns over time. It is typically more structured and relies on quantitative analysis. Prediction, on the other hand, can be based on intuition or qualitative assessments and may not always use data. Predictions can be less formal and more subjective, often focusing on specific outcomes [23]. Here's an overview of some common methods used for forecasting:

2.2.1 Simple Linear Regression

Simple linear regression is a foundational statistical technique used in forecasting to understand and predict the relationship between two continuous variables. It provides insights into how changes in one variable can influence changes in another, facilitating the projection of future trends based on historical data patterns. In forecasting, simple linear regression helps analysts and businesses make informed decisions by quantifying the direction and strength of relationships between variables. For example, in sales forecasting, simple linear regression could be employed to predict future sales volumes based on historical marketing expenditures, allowing companies to adjust their strategies accordingly. Its intuitive approach and ability to identify trends make simple linear regression a valuable tool for making projections and strategic planning in various fields [24].

2.2.2 Multiple Linear Regression

Multiple linear regression extends the principles of simple linear regression by incorporating multiple predictor variables to enhance forecasting accuracy. In forecasting, multiple linear regression enables analysts to assess how several factors simultaneously influence an outcome of interest. The model can provide a more comprehensive understanding of the drivers behind a forecasted outcome by considering multiple predictors such as economic indicators, demographic data, or marketing expenditures. For instance, in real estate forecasting, multiple linear regression might incorporate variables like location, property size, and local economic conditions to predict housing prices. This approach allows forecasters to account for the complexity of real-world scenarios and make more informed predictions based on a holistic view of contributing factors [25].

2.2.3 Logistic Regression

Logistic regression, a type of regression analysis suitable for forecasting, focuses on predicting binary outcomes based on a set of predictor variables. Unlike linear regression, which predicts continuous values, logistic regression models the probability of a binary event occurring, such as whether a customer will purchase a product or not. This makes it particularly useful in scenarios where understanding the likelihood of an event is crucial for decision-making. For example, in credit risk assessment, logistic regression can predict the probability of a borrower defaulting based on variables like credit score, income, and debt-to-income ratio. By estimating probabilities rather than exact outcomes, logistic regression provides valuable insights into the likelihood of specific events, enabling businesses to mitigate risks and optimize strategies in forecasting applications [26].

2.3 Probability Distributions

Probability distributions quantify outcomes' likelihood under specific conditions, defined by parameters like mean and variance. Common types include the Normal Distribution for symmetric data, Poisson for counts, Exponential for event timing, and Binomial for binary outcomes. They are pivotal in statistical modelling, informing predictions across diverse fields like finance and healthcare [27].

2.3.1 Binomial Distribution.

Binomial Distribution is defined for a fixed number of independent trials, each with the same success probability *p*. It gives the probability of observing *k* successes in *n* trials as:

$$P(X=k) = \binom{n}{k} p^k (1 - p)^{n-k}, \; k = 0,1,2, \ldots \tag{2.1}$$

- p is the probability of success on a single trial.
- (1−p) is the probability of failure on a single trial.
- n is the total number of trials.
- k is the number of successes.

.

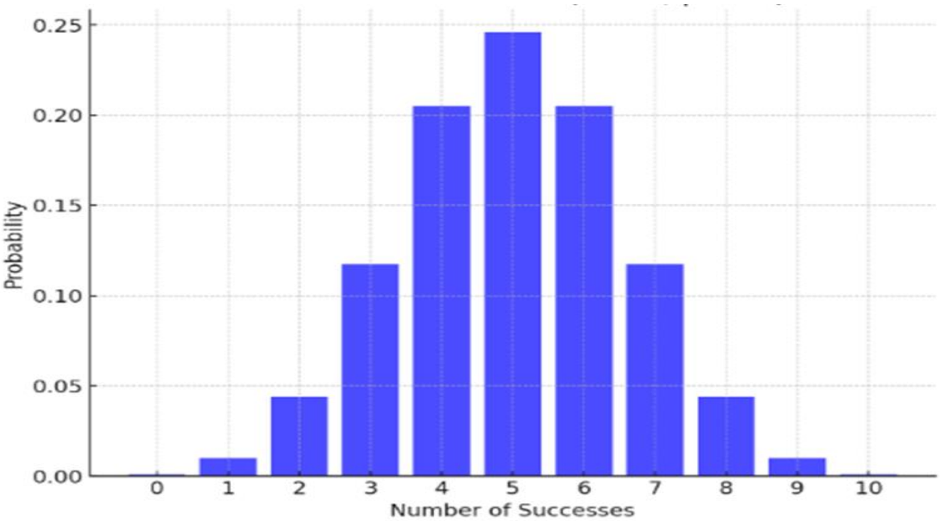

Figure 2. 1 The Binomial Distribution for n trials and a probability of success.

This formula succinctly captures the essence of binomial scenarios, facilitating the calculation of the likelihood of outcomes in experiments like coin tosses or binary outcomes in surveys [28].

Fig 2.1 represents a graph of the probability of achieving a specific number of successes (from 0 to 10) in these trials see Appendix 1. This visualization illustrates the distribution of probabilities across different possible outcomes in a binomial [29].

2.3.2 Poisson Distribution.

The Poisson Distribution models the probability of a certain number of events occurring within a fixed interval, assuming these events happen at a constant rate independently of each other. It is expressed as

$$P(X=k) = \frac{\lambda^k e^{-\lambda}}{k!} \quad , k = 0,1,2, \ldots.. \qquad (2.2)$$

where k represents the number of events, λ is the average rate of events per interval and e is the base of the natural logarithm.

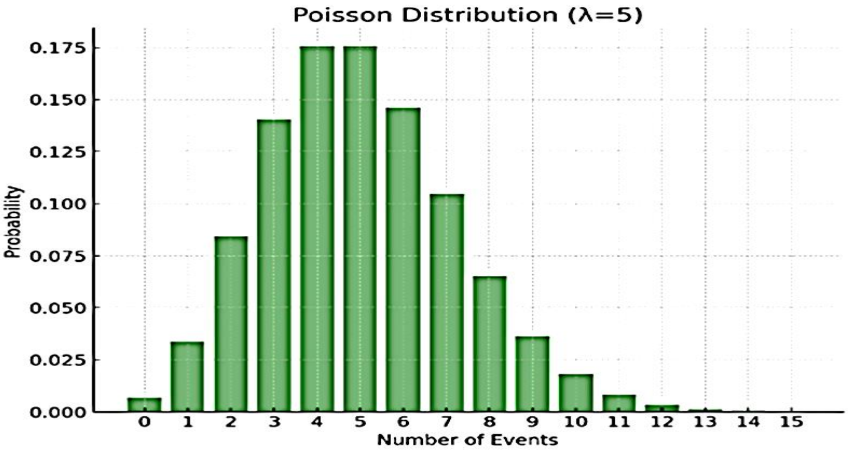

Figure 2.2 The Poisson Distribution with an average rate (λ) of 5.

Each bar in the graph represents the probability of observing a certain number of events (ranging from 0 to 15) in a given time interval as in Fig 2.2 see Appendix 2. This visualization provides insight into how the probabilities are distributed for different counts of events in a Poisson [30].

2.3.3 Geometric Distribution.

The geometric distribution is a discrete probability distribution that models the number of trials needed to achieve the first success in a sequence of independent Bernoulli trials, where each trial has only two possible outcomes: success (with probability p) or failure (with probability 1 - p).

The probability mass function (PMF) of the geometric distribution is given by:

$$P(X=k) = (1-p)^{k-1} \times p \qquad (2.3)$$

- P(X=k) is the probability that the first success occurs on the k-th trial.
- p is the probability of success on a single trial.
- (1−p) is the probability of failure on a single trial.
- k is the trial number at which the first success occurs (k=1,2,3,…)

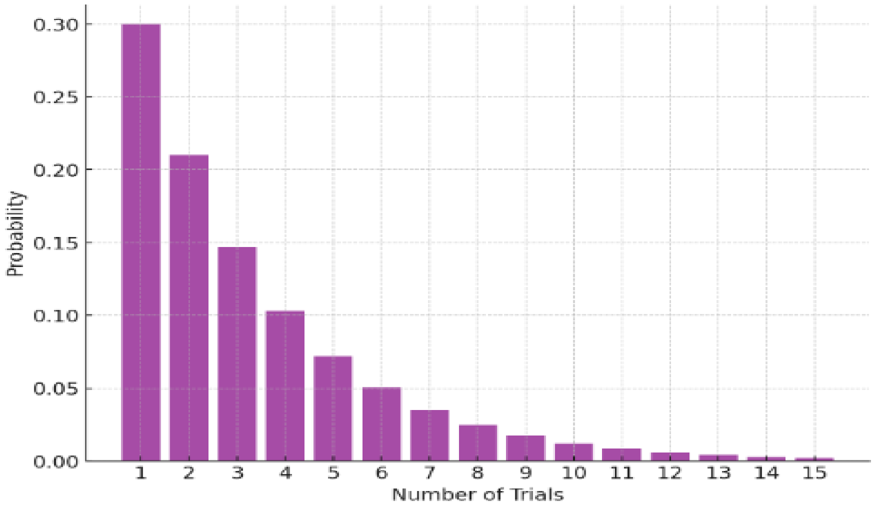

Figure 2.3 The geometric Distribution with a probability of success p=0.3

From Fig 2.3 each bar in the graph represents the probability of achieving the first success on a specific trial number (ranging from 1 to 15) see Appendix 3. This visualization illustrates how the probabilities decrease as the number of trials increases in a geometric distribution [31].

2.3.4 Hypergeometric Distribution.

The Hypergeometric Distribution is a crucial probability model used to compute the probability of achieving a specific number of successes in a fixed number of draws without replacement from a finite population. It is widely employed in fields such as manufacturing quality control and genetics research, where accurate prediction of sample outcomes is critical. For example, in quality control, it aids in estimating the likelihood of encountering a certain number of defective items in a sample drawn from a production batch. Similarly, in genetics, it helps predict the probability of observing a specified number of individuals with a particular genetic trait in a sample from a larger population. Unlike the Binomial Distribution, which assumes sampling with replacement, the Hypergeometric Distribution considers each draw affecting subsequent probabilities due to the finite nature of the population, making it ideal for scenarios requiring precise probability calculations for informed decision-making and analysis [32].

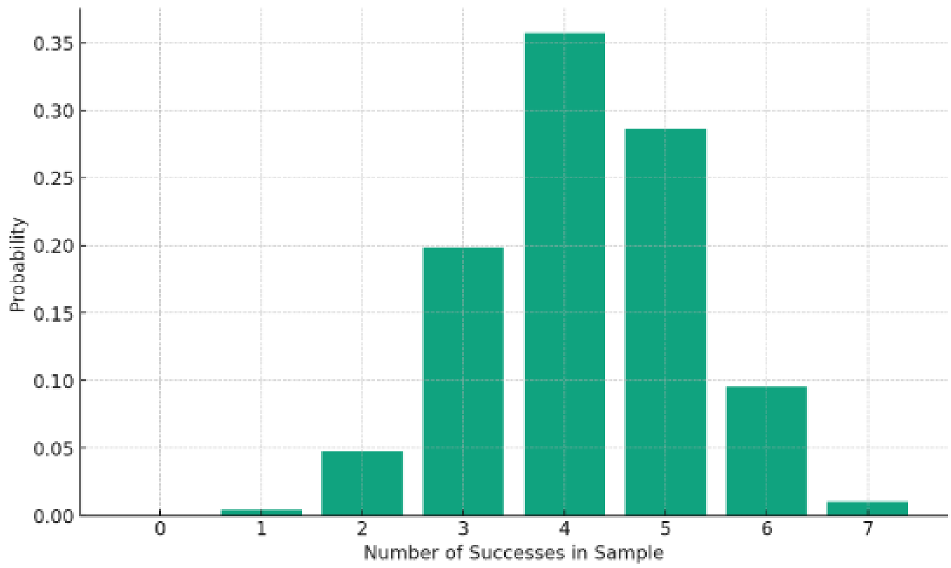

Figure 2.4 Hypergeometric Distribution.

The probability mass function (PMF) of the hypergeometric distribution is given by:

$$P(X=k) = \frac{\binom{K}{k}\binom{N-K}{n-k}}{\binom{N}{n}} \tag{2.4}$$

P(X=k) is the probability of drawing exactly k successes in n draws.

N is the total number of items in the population, n is the number of draws.

K is the total number of successes in the population.

Fig 2.4 shows The hypergeometric distribution is a probability distribution that describes the probability of a certain number of successes in a fixed number of draws, without replacement, from a finite population of two types of elements (e.g., successes and failures) see Appendix 4.

2.3.5 Normal (Gaussian) Distribution.

The normal distribution, also known as the Gaussian distribution, is a continuous probability distribution that is symmetric about its mean, forming a bell-shaped curve as in Fig 2.5 see Appendix 5. It is characterized by two parameters: the mean (μ) and the standard deviation (σ) [33].

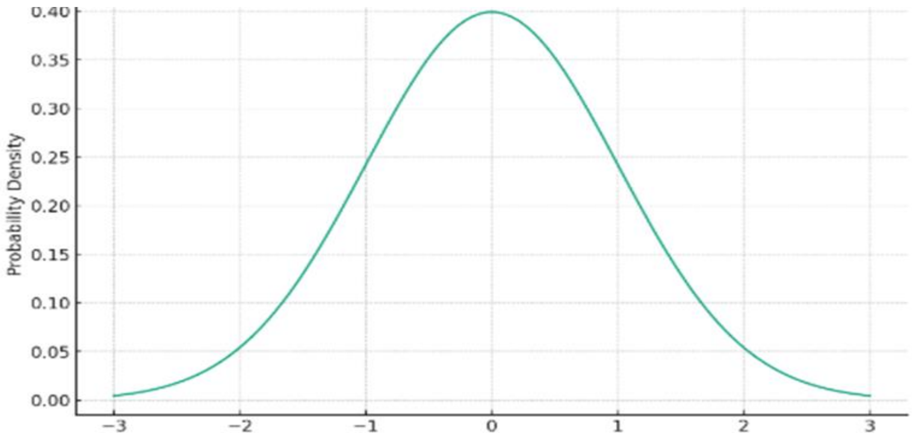

Figure 2.5 The Normal (Gaussian) Distribution.

Probability density function (PDF): The probability density function of the normal distribution is given by:

$$f(x|\mu,\sigma^2)=\frac{1}{\sqrt{2\pi\sigma^2}}\,e^{-\left(\frac{(x-\mu)^2}{2\sigma^2}\right)} \tag{2.5}$$

Where:

x is a random variable, μ is the mean of the distribution., σ^2 is the variance of the distribution.

2.3.6 Exponential Distribution.

The exponential distribution is a continuous probability distribution that describes the time between events in a Poisson process, where events occur continuously and independently at a constant average rate. It is characterized by a single parameter, often denoted as λ (lambda), which represents the rate at which events occur [34].

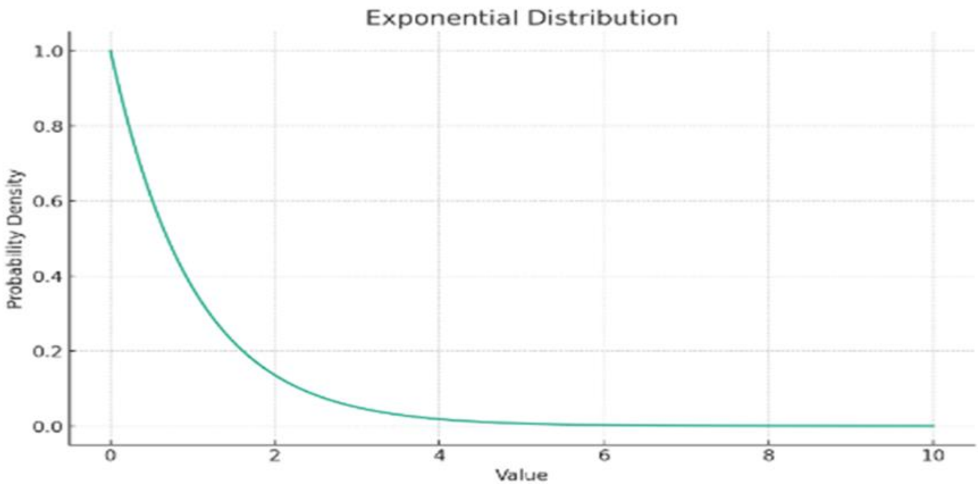

Figure 2.6 The Exponential Distribution.

The rate parameter (λ) is set to 1. Fig 2.6 shows the probability density of different values see Appendix 6, illustrating the characteristic rapid decrease of the exponential distribution, where higher values become increasingly less likely [35]. Probability density function (PDF): The probability density function of the exponential distribution is given by:

$$f(x) = \lambda e^{-\lambda x} \quad \text{for } x \geq 0 \tag{2.6}$$

- λ (lambda) is the rate parameter. It represents the average number of events that occur in a unit of time.

2.3.7 Uniform Distribution.

The uniform distribution is a continuous probability distribution where every value within a certain range has an equal probability of occurring. It is often denoted as $U(a,b)$, where a and b are the parameters representing the lower and upper bounds of the range, respectively[36].

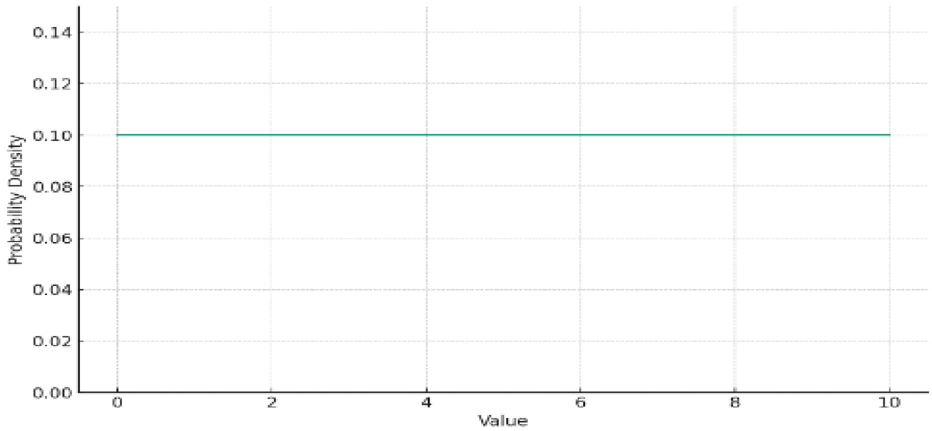

Figure 2.7 The Uniform Distribution.

Fig 2.7 shows the Uniform Distribution See Appendix 7. The probability density function of the uniform distribution over the interval $[a,b]$ is given by:

$$f(x) = \frac{1}{b-a} \quad \text{for} \quad a \leq x \leq b \tag{2.7}$$

This means that the probability of any value within the interval $[a,b]$ is constant and equal to $\frac{1}{b-a}$. Outside of this interval, the probability density is zero [37]

2.3.8 Beta Distribution.

The beta distribution is a continuous probability distribution defined on the interval [0,1]. It is a versatile distribution commonly used in Bayesian statistics, machine learning, and modeling random variables that represent proportions or probabilities [38].

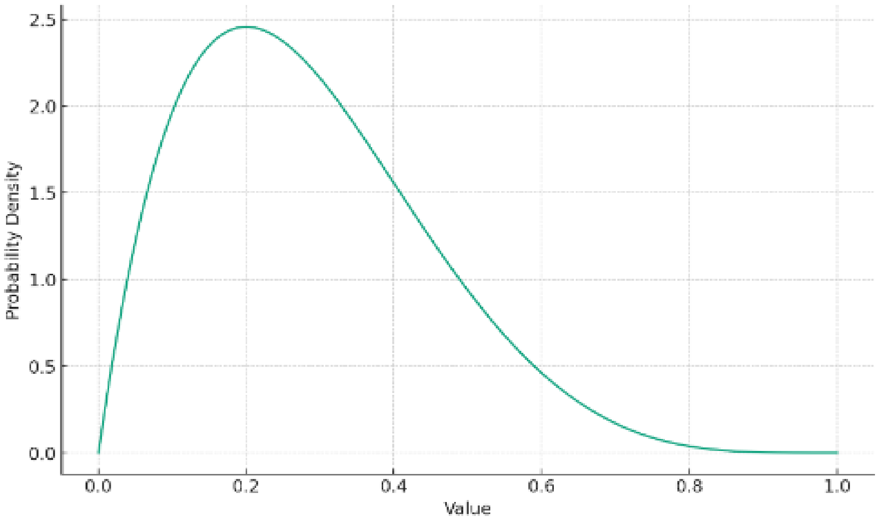

Figure 2.8 The Beta Distribution.

Fig 2.8 shows The Beta Distribution See Appendix 8. The probability density function of the beta distribution is given by:

$$f(x) = \frac{x^{\alpha-1}(1-x)^{\beta-1}}{B(\alpha,\beta)} \quad (2.8)$$

- α and β are shape parameters, with $\alpha,\beta > 0$.
- $B(\alpha,\beta)$ is the beta function, which is a normalization constant that ensures the integral of the PDF over the interval [0,1] equals 1[39].

2.4 Related work.

Predictive modeling, a cornerstone of data science, has revolutionized our ability to anticipate future events by analyzing historical data. Its applications are diverse, spanning finance, healthcare, meteorology, and more. This exploration delves into the related work across all methods of prediction, underscoring the evolution and diversity of these techniques [40].

The roots of predictive modeling are ancient, initially based on numerical forecasting and simple mathematical models derived from astronomical observations. These early methods laid the groundwork for the sophisticated predictive models we see today. The advent of statistical science marked a turning point, formalizing predictive modeling with techniques like linear regression. These early statistical methods allowed for predictions based on the relationships between variables, fundamentally changing fields such as economics and agriculture [41].

As computational power surged in the late 20th century, machine learning algorithms emerged, capable of identifying complex patterns in data. This period marked a significant advancement in predictive modelling, enabling more nuanced predictions across various domains. The introduction of deep learning, a subset of machine learning that utilizes neural networks, further enhanced the ability to make predictions from high-dimensional data, revolutionizing fields like image and speech recognition [42].

Time series analysis and Bayesian methods have also played pivotal roles in the evolution of predictive modeling. Time series analysis, crucial in economics and weather forecasting, analyzes data points collected over time to predict future values. Bayesian methods, which provide a probabilistic approach to prediction, have been instrumental in drug development and epidemiological forecasting, allowing for the integration of prior knowledge with new evidence [43].

Ensemble methods like Random Forest and Gradient Boosting combine models for enhanced accuracy. Markov models predict future states from current conditions, while HMMs handle scenarios where states are indirectly observed, used in diverse fields like natural language processing and bioinformatics [44].

Predictive modeling's application in weather forecasting and the management of COVID-19 has been particularly noteworthy. In meteorology, models like time series analysis, Markov models, and ensemble methods have been employed to predict weather patterns and atmospheric conditions. These models, by analyzing sequences of meteorological data, have significantly improved the accuracy and reliability of weather forecasts, aiding in disaster preparedness and agricultural planning [45].

The COVID-19 pandemic underscored the importance of predictive modeling in public health. Machine learning algorithms, time series analysis, and Bayesian methods have been used to forecast the spread of the virus, anticipate outbreaks, and assess the impact on healthcare systems. These predictive models have been crucial for public health officials in planning interventions, allocating resources, and implementing measures to control the spread of the virus [46].

In this thesis, we explore the utilization of Markov Models and HMMs in the prediction of weather patterns and the spread of COVID-19, highlighting the versatility and effectiveness of these models in handling time-series data and processes with inherent uncertainties. Markov Models, known for their principle of memory lessness, provide a robust framework for predicting future states based solely on the current state of a system. This property is particularly advantageous in modeling weather systems, where the future atmospheric condition can often be approximated as dependent primarily on the current state, despite the underlying complexity and dynamism of weather patterns. Through the application of Markov Models, this thesis aims to demonstrate the potential for improved accuracy in short-term weather forecasting, facilitating better preparedness for meteorological events. HMMs analyze systems where states are indirectly observed via emissions sequences. They are crucial for modeling complex dynamics like COVID-19 spread, and inferring infection trends from observable data (e.g., reported cases). This thesis enhances pandemic prediction and resource allocation in public health [47].

The thesis methodology involves collecting and preprocessing meteorological and epidemiological datasets. For weather prediction, a Markov Model learns transition probabilities between weather conditions using historical data, tested for accuracy on new data. For COVID-19 spread analysis, an HMM models pandemic stages based on reported cases and testing rates, assessing its ability to forecast infection trends and intervention effects [48].

The thesis highlights Markov Models and HMMs as effective tools for forecasting weather and COVID-19 spread. Markov Models excel in short-term weather predictions, while HMMs provide detailed insights into pandemic dynamics, including transmission probabilities and intervention impacts. These models enhance prediction accuracy in meteorology and public health decision-making [49].

Researchers use Markov models and Hidden Markov Models (HMMs) to assess banknote authenticity and value. Markov models analyze the likelihood of different banknote values and their authenticity. HMMs track potential transformations of banknotes after use. These models help researchers understand currency dynamics, aiding in informed decisions about money management and counterfeit prevention [50].

Chapter 3

Forecasting using Markov model and its applications.

Chapter (3) Forecasting using Markov model and its applications.

3.1 Fundamentals of Markov Chains

3.1.1 State Space

A Markov chain consists of a set of states, representing all possible conditions or situations of the system being modeled. These states can be discrete or continuous.

Let $S=\{S_1,S_2,\ldots,S_n\}$ be the set of states in the Markov chain. A Markov chain is a stochastic process that exhibits the Markov property, which states that the future state depends only on the present state and not on the sequence of events that preceded it. Formally, let X_t denote the state of the system at time t. A Markov chain satisfies the Markov property if for any time t and states $i_0,i_1,...,i_t$ in the state space S, the probability of transitioning to state i_{t+1} from state i_t depends only on the current state i_t [51].

This property is expressed mathematically as:

$$P(X_{t+1}=i_{t+1}|X_t=i_t,X_{t-1}=i_{t-1},...,X_0=i_0)=P(X_{t+1}=i_{t+1}|X_t=i_t). \tag{3.1}$$

Markov chains have several basic properties. One such property is the transition probability matrix P, where P_{ij} represents the probability of transitioning from state i to state j. Another fundamental concept is the notion of a stationary distribution, where a distribution π is stationary if $\pi P=\pi$, implying that the distribution remains unchanged over time. These properties form the foundation for analyzing the behavior and long-term dynamics of Markov chains in various applications, including modeling random processes in physics, biology, and economics [52].

3.1.2 Memoryless Processes:

Memoryless processes, also known as Markov processes, are stochastic processes in which the future behavior of the process depends only on its current state and not on its history. This characteristic is expressed mathematically through the memoryless property, which states that the conditional probability distribution of future states given the present state is independent of past states. Formally, for a process X_t with memoryless property, the conditional probability $P(X_{t+s}|X_t)$ is equal to $P(X_s)$, where s represents the time interval between the present and future states [53].

$$P(X_{t+s}|X_t)=P(X_s). \tag{3.2}$$

Memoryless processes find application in various fields, including queuing theory, reliability engineering, and telecommunications. Exponential distributions often model memoryless processes, where the time until the next event follows an exponential distribution with a constant rate parameter. This property simplifies the analysis and allows for efficient modeling of systems where history does not influence future outcomes, making memoryless processes a valuable tool in probability theory and applied mathematics [54].

3.2 Transition Probabilities.

3.2.1 Transition Matrices.

Constructing represents the probability of transitioning from state s_i to state s_j. These probabilities must satisfy certain conditions, such as being non-negative and summing to one along each row, ensuring that the process moves from one state to another with well-defined probabilities. Associated with each pair of states is a transition probability, which indicates the likelihood of transitioning from one state to another in the next time step. These probabilities are often represented in the form of a transition matrix or transition diagram [55].

In a discrete-state Markov chain, the transition probabilities are typically organized into a square matrix called the transition matrix. Each element of the matrix represents the probability of transitioning from one state to another. Let P=[p_{ij}] be the transition probability matrix, where p_{ij} represents the probability of transitioning from state S_i to state S_j [56]. The transition matrix P satisfies:

- $p_{ij} = \begin{pmatrix} p_{11} & p_{12} & \cdots & p_{1j} \\ p_{21} & p_{22} & \cdots & p_{2j} \\ \vdots & \vdots & \ddots & \vdots \\ p_{i1} & p_{i2} & \cdots & p_{ij} \end{pmatrix}$ (3.3)
- $0 \leq p_{ij} \leq 1$, for all i,j=1,2,…,n. (3.4)
- $\sum_{j=1}^{n} p_{ij}$ =1, for all i=1,2,…,n. (3.5)

`

3.2.2 Initial Distribution

The initial distribution in a Markov chain represents the probability distribution over states at the beginning of the chain. It is represented as

π_0=[$\pi_0(1)$,$\pi_0(2)$,…,$\pi_0(n)$]. (3.6)

where:

- $\pi_0(i)$ represents the probability of starting in state S_i.
- $\pi_0(i) \geq 0$ for all i=1,2,…,n.
- $\sum_{i=1}^{n} \pi_0(i)$=1, ensuring that the probabilities sum up to 1, indicating that the system starts in one of the states [57].

In other words, the initial distribution specifies the likelihood of the system being in each state at the initial time step. For example, if we have a simple Markov chain with three states S_1, S_2, and S_3, the initial distribution might be $\pi_0 = [0.3 \quad 0.4 \quad 0.3]$. This means that there is a 30% chance of starting in state S_1, a 40% chance of starting in state S_2, and a 30% chance of starting in state S_3.

The initial distribution is essential because it provides the starting point for analyzing the evolution of the Markov chain over time. It influences the probabilities of transitioning to different states in the next time step according to the transition probabilities defined by the transition matrix P, another example [58]. See Appendix 9

Once the transition matrix is constructed, it becomes a powerful tool for analyzing the behavior of the underlying Markov chain. By manipulating the transition matrix, various properties and characteristics of the Markov chain can be studied, such as the long-term behavior, steady-state distributions, and expected time to reach certain states. Additionally, techniques such as matrix multiplication and eigenvalue analysis enable the prediction of future states and the exploration of the system's dynamics over time. Thus, constructing transition matrices serves as a crucial step in understanding and utilizing Markov chains for modelling real-world processes and systems [59].

3.3 Properties of States.

In the realm of Markov chains, states can be classified based on their properties, shedding light on the dynamics and behaviour of the system. One common classification scheme distinguishes between transient and recurrent states. A state i is transient if, starting from that state, there is a non-zero probability of never returning to it. Mathematically, for a state i to be transient, the probability of eventually returning to it, denoted p_{ii}, is zero. Conversely, a state is recurrent if it is certain to be visited again eventually, meaning $p_{ii}=1$.

Recurrent states can further be categorized into two types, positive recurrent states, where the expected return time is finite, and null recurrent states, where the expected return time is infinite [60].

Another important property of states in Markov chains is the notion of communicating classes. States i and j are said to communicate with each other if there exists a sequence of states that leads from i to j with positive probability and vice versa. Communicating classes partition the state space into disjoint sets, where all states within a class communicate with each other but not with states outside the class. Mathematically, a communicating class C satisfies the property that for any i,j in C, both i communicates with j and j communicates with i. Understanding these classifications and properties provides valuable insights into the long-term behavior and convergence properties of Markov chains, aiding in the analysis and prediction of various stochastic processes[61].

3.4 Steady-State and Limiting Distributions.

In the study of Markov chains, steady-state distributions offer essential insights into the long-term behavior of systems. A steady-state distribution represents the equilibrium distribution of probabilities over the states of the system, indicating the proportion of time the system spends in each state as time approaches infinity. Mathematically, for a Markov chain with transition matrix P, the steady-state distribution π satisfies the equation $\pi=\pi \cdot P$, where π is a probability vector. This equation implies that the distribution remains unchanged from one-time step to the next, representing a state of balance in the system where the probabilities of transitioning between states stabilize over time [62].

The concept of limiting distributions further extends the understanding of steady-state behavior in Markov chains. In cases where the initial distribution of states is not the steady state, but the system evolves over time, the limiting distribution represents the distribution of states as time approaches infinity. Mathematically, if X_t denotes the state vector at time t, and $\lim_{t\to\infty} X_t = \pi$, then π is the limiting distribution of the Markov chain [63].

Analyzing steady-state and limiting distributions provides valuable insights into the behaviour and convergence properties of Markov chains, aiding in the prediction and understanding of various stochastic processes in fields such as economics, biology, and engineering [64].

3.5 Absorbing States and Markov Chains.

Absorbing states are pivotal in Markov chain theory, acting as endpoints where transitions cease, affecting models in epidemiology, queueing systems, and biochemical kinetics. For instance, in disease spread models, they denote recovery or mortality states, shaping how infections propagate. Analyzing Markov chains with absorbing states focuses on crucial metrics like expected time until absorption, which estimates average steps to reach an absorbing state, and probability of absorption, indicating the likelihood of entry into these states. Techniques such as fundamental matrix computation and Monte Carlo simulations explore these dynamics, offering insights into long-term system behavior [65].

3.6 Ergodic Theorems in Markov Chains.

Ergodic theorems in the context of Markov Chains are pivotal in understanding their long-term behavior and statistical properties. A key theorem is the Ergodic Theorem for Markov Chains, which asserts that for an ergodic Markov Chain with a stationary distribution π, the time-average of a function f along the chain's trajectory converges to its expected value under π.

$$\lim_{n\to\infty} \frac{1}{n} \sum_{k=1}^{n} f(X_k) = \sum_k \pi(x) f(x). \quad (3.7)$$

Here, X_k denotes the state of the Markov Chain at the time k , $\pi(x)$ is the probability of state x under the stationary distribution, and the summation is over all possible states x. This theorem underpins the predictability of the chain's long-term behaviour based on its stationary properties.

Moreover, the Law of Large Numbers for Markov Chains further solidifies these insights, stating that under ergodicity, the sample average of $f(x)$ converges almost surely to the expected value $\sum_k \pi(x) f(x)$:

$$\frac{1}{n}\sum_{k=1}^{n} f(X_k) \rightarrow \sum_k \pi(x) f(x) \quad as\, n \rightarrow \infty$$

This law ensures that as the number of steps n increases, the empirical average of f along the chain approaches the expected value of f to the stationary distribution π, providing a fundamental link between theoretical predictions and empirical observations in Markov Chain analysis.

These theorems are fundamental in statistical inference and decision-making processes that rely on Markov Chains, offering rigorous mathematical foundations for understanding the stability and convergence of such stochastic processes over time [66].

The most common ergodic theorem in Markov chain theory is the Ergodic Theorem for Homogeneous Markov Chains. Let P be the transition matrix of a homogeneous Markov chain with a finite state space S. If the chain is irreducible and aperiodic, then:

1. There exists a unique stationary distribution π such that $\pi P = \pi$.

2. For any initial distribution μ, the distribution of the chain converges to π as n tends to infinity, i.e., $\lim_{n\to\infty} \mu P^n = \pi$ [67].

Irreducibility ensures that every state can be reached from any other state with positive probability, and aperiodicity guarantees that there are no periodicities in the chain that prevent it from converging to a stationary distribution. The stationary distribution π represents the long-term proportion of time the chain spends in each state. It is also the unique left eigenvector of P corresponding to the eigenvalue 1. Ergodic theorems are essential for understanding the behavior of Markov chains over time and for making predictions about their long-term properties. They find applications in various fields, including statistical mechanics, queueing theory, and machine learning [68]. See Appendix 10

3.7 Maximum Likelihood Estimation

Maximum Likelihood Estimation (MLE) for Markov Chains typically involves estimating the parameters of the transition matrix based on observed sequences of states. Here's a step-by-step outline of how MLE can be applied to estimate parameters for a Markov Chain:

Let's consider a discrete-time homogeneous Markov Chain with $\boldsymbol{N}$ states

$\{S_1, S_2, \ldots, S_N\}$.

Transition Matrix P: This N×N matrix $P=[p_{ij}]$ represents the probabilities of transitioning from state S_i to state S_j. Each row i sums to 1.

Maximum Likelihood Estimation (MLE):To estimate the transition probabilities p_{ij}, we can use the observed sequences of states.

3.7.1 Data Preparation:

Obtain a sequence of observations $X = (X_1, X_2, \ldots, X_t)$, where X_t represents the state observed at time t.

3.7.2 Likelihood Function:

The likelihood function $\mathrm{L}(\mathrm{P};\mathrm{X})$ represents the probability of observing the sequence X given the transition matrix P.

For a Markov Chain, assuming the initial state distribution π (where π_i is the probability of starting in state S_i), the likelihood function is:

$$\mathrm{L}(\mathrm{P};\mathrm{X}) = \pi(X_1)\prod_{t=2}^{T} pX_{t-1}, X_t \tag{3.8}$$

Here, $\pi(X_1)$ is the probability of starting in the state X_1 , and pX_{t-1}, X_t is the transition probability from state X_{t-1} to state X_t.

3.7.3Log-Likelihood Function:

To simplify computations and handle numerical stability, we often use the log-likelihood function:

$$\ell(\mathrm{P};\mathrm{X}) = \log\mathrm{L}(\mathrm{P};\mathrm{X}) = \log\pi(X_1) + \prod_{t=2}^{T}\log\ pX_{t-1}, X_t \tag{3.9}$$

MLE Estimation: Find the maximum likelihood estimate $\hat{P}$ of the matrix P

By maximizing $\ell(\mathrm{P};\mathrm{X})$ with respect to P.

3.8 Application of weather forecasting using the Markov model.

3.8.1 Markov model.

Creating a Markov model involves several steps, particularly when dealing with a sequence of events like weather states that evolve. The goal is to estimate the probability of transitioning from one state to another, which in the context of weather, translates into predicting future weather based on current conditions. Here's how we can approach building a Markov model and its associated transition matrix for the weather data:

The states in our Markov model will be the different types of weather conditions (like "rain", "sunny", "drizzle", etc.). Each state represents a possible condition on any given day. For a Markov model, we need sequential data. In this case, our data sequence is the daily weather conditions. This data must be sorted by date without any gaps for accurate analysis. We need to count how often each type of weather transitions to another type. This involves tallying occurrences where one type of weather follows another from day to day.

The transition matrix is constructed from the counts of transitions between states. Each element in the matrix a_{ij} represents the probability of moving from state i to state j. The probabilities are calculated by dividing the count of transitions from i to j by the total transitions from i. The weather states in the dataset are 'rain', 'sun', 'fog', 'drizzle', and 'snow'. Now, let's calculate the transition probabilities between these states.

Table 3.1 shows the transition matrix for weather states, representing the probabilities of transitioning from one weather state to another based on the provided data.

Table 3.1 The transition matrix for weather states.

	Drizzle	Rain	Sun	Snow	Fog
Drizzle	0.308	0.346	0.308	0.019	0.019
Rain	0.030	0.674	0.231	0.016	0.050
Sun	0.023	0.225	0.681	0.008	0.063
Snow	0.00	0.423	0.192	0.385	0.00
Fog	0.030	0.356	0.337	0.00	0.277

Each row in the matrix represents the probabilities of transitioning from the weather state labelled on the left ('From') to the states across the top ('To'). For example, if it's raining today, there's a 67.4% chance it will rain again tomorrow, a 23.1% chance it will be sunny, and so on. This matrix can be useful for predicting weather changes based on historical data.

To ensure that each row of the transition matrix sums to 1 (which is necessary because these rows represent probability distributions), we normalize the matrix by row.

3.8.2 Explanation.

- Rows represent the weather state of the previous day.
- Columns represent the weather state of the current day.
- Values in the matrix represent the probability of transitioning from the state in the row to the state in the column from one day to the next.

3.8.3 Transition Matrix.

This matrix can be used for stochastic weather predictions where the next day's weather state is predicted based on the current day's state, using the probabilities from the matrix. For example, if it is "sunny" today, the matrix provides the probabilities for each type of weather tomorrow. An initial probability matrix, also known as the initial state distribution, defines the likelihood of starting in each state when observing a sequence. For the case of weather states, this matrix will indicate the probability of each type of weather being the starting state on the first day of the observations. To define this initial matrix from your dataset, we'll calculate the probability of each weather state occurring on the first day recorded in the dataset. Here’s how we’ll do it:

1. Identify the weather state on the first day of each recorded sequence. If your dataset represents continuous daily records without breaks, we'll just use the first entry.

2. Calculate the proportion of times each weather state appears as the first state relative to the total number of sequences.

Since your dataset seems to represent continuous records, we'll proceed with determining the weather state on the first day of the dataset and treat it as the initial state. This approach simplifies the process under the assumption that the dataset contains one continuous sequence. Let's calculate this: The initial state of the weather in your dataset is 'drizzle'. This implies that on the first day of the recorded data, the weather was drizzling. If we were to express this in terms of an initial probability matrix for a more extended dataset or multiple sequences, we would calculate the proportion of times each state occurs as the first state and present it as follows, for example:

Table 3.2 The initial state of the weather in your dataset is 'drizzle'.

State	Probability
Drizzle	1.0
Rain	0.0
Sun	0.0
Snow	0.0
Fog	0.0

Table 3.2 shows the matrix shows that there is a 100% probability that the sequence starts with 'drizzle', based on the first day's data. To predict the weather after 5 days using the transition matrix, we'll repeatedly apply the transition matrix to the initial state vector. This process, known as computing

the power of the transition matrix, gives us a probability distribution of the weather states after a certain number of steps. Here's how you can think about it:

1. Start with the initial state vector, where the state is 'drizzle'.
2. Apply the transition matrix once to get the probability distribution of the weather on the next day.
3. Repeat applying the transition matrix to the result from the previous day for a total of 5 times.

Since 'drizzle' is the initial state, the initial state vector would be:

Initial vector = [1,0,0,0,0] This corresponds to the states ['Drizzle', 'Rain', 'Sun', 'Snow', 'Fog']. After applying a transition matrix to predict weather states over 5 days starting from an initial state of 'drizzle', the calculated probabilities for each state are as follows: 'Drizzle' has a 4.32% likelihood, 'Rain' is predicted at 33.21%, 'Sun' at 42.59%, 'Snow' at 5.77%, and 'Fog' at 14.11%. These probabilities indicate that 'Sun' is the most probable weather state after 5 days, followed by 'Rain'. This matrix-based prediction offers valuable statistical insights into the expected evolution of weather conditions based on historical transition patterns, helping to anticipate likely weather outcomes over the specified time.

3.8.4 Results.

After analyzing Seattle's weather dataset, key insights were gleaned using a Transition Matrix. The matrix computed probabilities of transitioning between weather states, highlighting high probabilities for 'rain' (67.4%) and 'sun' (68.1%), indicating their tendency to persist day-to-day. Beginning with the initial state of 'drizzle', derived from the dataset's starting point, we applied the matrix iteratively to predict weather conditions five days ahead. The forecast indicated 'Sun' as the most likely state (42.59%), followed by 'Rain' (33.21%), underscoring the predictive power of transition matrices in forecasting future weather patterns based on historical observations.

3.8.5 Conclusion

The transition matrix approach proved invaluable in forecasting future weather conditions based on historical data. The analysis uncovered several key insights. Firstly, it identified 'Sun' and 'Rain' as the predominant consecutive weather states, indicating a dynamic yet somewhat predictable pattern shaped by these conditions. Secondly, the method demonstrated predictive capabilities, notably forecasting a high likelihood of sunny weather following five days starting from a drizzly day, underscoring the transition matrix's usefulness in providing statistical forecasts for planning and preparation purposes. However, the model's limitations were also recognized; it assumes transitions are solely influenced by the previous day's state, neglecting other potentially influential factors such as seasonal variations, geographic influences, or extreme weather events. This case study serves as a prime example of how statistical models can extract meaningful predictions and insights from real-world data, facilitating a better understanding and forecasting based on historical patterns.

3.9 Prediction of COVID-19 Using a Markov Model.

The Markov model constructed for confirmed COVID-19 cases in Bangladesh uses a state-based approach to predict transitions between different levels of confirmed cases categorized as Low, Medium, and High. The model is based on transition probabilities that indicate the likelihood of moving from one state to another on consecutive days. These probabilities are derived from historical data and capture the underlying dynamics of the pandemic’s progression.

3.9.1 Markov model.

The model uses a matrix to understand how COVID-19 cases change over time. Each row in the matrix shows if cases are low, medium, or high now. Each column shows what happens next. For example, if cases are low today, there's a 91.7% chance they stay low tomorrow. But there are risks: an 8.3% chance they go from low to medium, and a 7.7% chance they go from medium to high. Once cases are high, there's a 100% chance they stay high. There's no chance they go back down without help. This shows how hard it is to lower case numbers without acting. Understanding these probabilities helps us plan how to stop the virus from spreading and manage outbreaks better.

3.9.2 Explanation.

The model's predictive power enables forecasting the probable future state of confirmed COVID-19 cases using current data, offering valuable insights for health authorities to strategize resource allocation and plan effectively. Moreover, it aids in comprehending the dynamics of the disease's progression over time, facilitating the simulation of diverse scenarios based on distinct intervention strategies or shifts in public behaviour. Additionally, policymakers can leverage the analysis of transition probabilities to grasp the potential consequences of their decisions, thus enhancing informed policy-making processes and anticipating various outcomes more accurately.

3.9.3 Transition Matrix.

Let's present the transition matrix as a structured table, Table 3.3 represents a Markov transition matrix. This matrix will display the transition probabilities between the states of "Low", "Medium", and "High" for daily new COVID-19 cases:

Table 3.3 Probability matrix for confirmed COVID-19 cases in Bangladesh.

	Low	Medium	High
Low	0.63	0.31	0.06
Medium	0.44	0.45	0.11
High	0.09	0	0.91

To create an initial probability matrix for the states Low, Medium, and High with the specified probabilities as in Table 3.4, we can present this as a single-row matrix or table. This initial distribution reflects the initial likelihood of being in each of these states:

Table 3.4 The initial probability matrix for the states Low, Medium, and High.

State	Low	Medium	High
Probability	0.1	0.5	0.4

This initial probability matrix provides a starting point for the system, indicating that initially, there is a 10% chance of being in the Low state, a 50% chance of being in the Medium state, and a 40% chance of being in the High state. To find the probability distribution of the states (Low, Medium, High) after 10 days, given the initial distribution and the transition matrix, we can use matrix multiplication. Each multiplication of the current distribution by the transition matrix gives us the next day's distribution.

We'll perform this operation 10 times, starting from the initial distribution. Here are the steps:

1. Define the initial distribution vector.
2. Define the transition matrix based on the probabilities provided earlier.
3. Multiply the initial distribution by the transition matrix repeatedly for 10 iterations.

After 10 days, the probability distribution for the states Low, Medium, and High, based on the initial distribution and transitions as in Table 3.5

Table 3.5 After 10 days, the probability distribution for the states is Low, Medium, and High.

State	Probability
Low	0.34
Medium	0.20
High	0.46

3.9.4 Results.

The analysis of COVID-19 case outcomes over 10 days reveals varying probabilities for different levels: a significant likelihood (0.34) of cases decreasing to low levels suggests potential improvement from higher initial counts. Medium case levels decrease to 0.2, down from an initial 50%, indicating a moderate reduction. Conversely, the High state remains high at 0.46, implying sustained or increased case numbers due to ongoing transmission. These probabilities highlight the dynamic nature of disease spread, influenced by initial conditions and daily transitions. The emphasis on the High state underscores the importance of sustained public health measures to effectively control transmission.

3.9.5 Conclusion.

The Markov model demonstrated that the state of "High" cases is very stable with a high likelihood of remaining High (90.91%) and becomes the most probable state after 10 days (46.11%). This suggests that once the number of cases escalates, it is likely to remain high without effective intervention. The probabilities for Low and Medium also indicate possible transitions but show less stability compared to the High state.

Chapter 4

Forecasting Using Markov Model and Hidden Markov Model for the States of Banknotes.

Chapter (4) Forecasting Using Markov Model and Hidden Markov Model for the States of Banknotes.

4.1 Introduction to HMMs.

Hidden Markov Models (HMMs) are statistical models that represent systems composed of a series of unobservable (hidden) states. They are a powerful tool in various fields such as speech recognition, natural language processing, bioinformatics, and finance, among others. The core idea behind HMMs is to model a system where the observed data is generated by some hidden process that can be in one of several states. The model captures the probabilities of transitioning between these hidden states and the probabilities of generating observed data from each state [69].

4.1.1 states

Let $S=\{S_1,S_2,...,S_N\}$ be the set of N hidden states in the model. states refer to the internal configurations or conditions of the system being modelled, which are not directly observable from the outside. These states are "hidden" in the sense that while they determine the behaviour of the system and the likelihood of various observable outcomes, they cannot be directly measured or seen. The only evidence of these states comes through the observations that are related to these states through a probability model [70].

4.1.2 Observations

Let $V=\{v_1,v_2,...,v_M\}$ be the set of M possible observations, observations refer to the sequence of data or signals that are directly measurable or visible, which arise as a result of the system transitioning through a series of hidden states. Each observation is associated with a particular state, but because the states are hidden, the exact state that generated the observation is not directly known. Instead, the relationship between states and observations is governed by probability rules defined within the HMM framework [71].

4.1.3 Transition Probability.

Transition probabilities in Hidden Markov Models (HMMs) define the likelihood of moving from one hidden state to another in the model. These probabilities are crucial for understanding the dynamics of the system being modelled, as they capture the structure and behaviour of the process over time [72].

Given a set of N hidden states $S=\{S_1,S_2,...,S_N\}$, the transition probabilities are represented by a matrix A, where each element a_{ij} within this matrix represents the probability of transitioning from state S_i to state S_j. Formally, the transition probability a_{ij} is defined as:

$$a_{ij}=P(q_{t+1}=S_j|q_t=S_i)$$

- q_t is the state at time t,
- q_{t+1} is the state at time $t+1$,
- a_{ij} is the probability of transitioning from state i to state j.

For all i and j, transition probabilities must satisfy the following constraints:

1. Non-negativity: $a_{ij} \geq 0$, meaning the probability of transitioning from any state to any other state (including itself) must be non-negative.
2. Normalization: $\sum_{j=1}^{N} a_{ij}=1$ for all i, ensuring that the probabilities of moving from any given state to all possible next states sum up to 1.

The transition probabilities are often represented in matrix form, with matrix A being an $N \times N$ matrix where N is the number of states in the model [73].

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1N} \\ a_{21} & a_{22} & \cdots & a_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ a_{N1} & a_{N2} & \cdots & a_{NN} \end{pmatrix} \tag{4.1}$$

In HMMs, the transition matrix A is a fundamental component that, along with the observation emission probabilities and initial state probabilities, defines the behavior of the model. It is used in various algorithms associated with HMMs.

1. The Forward algorithm for computing the likelihood of an observed sequence.
2. The Viterbi algorithm for determining the most likely sequence of hidden states.
3. The Baum-Welch algorithm for learning model parameters from observed data.

Transition probabilities are a key feature of HMMs that enable the model to capture the temporal dynamics of sequences where the underlying state process is hidden and only observable through emitted symbols or signals [74]. Figure 4.1 shows the Hidden Markov Model with three possible emissions and two hidden states.

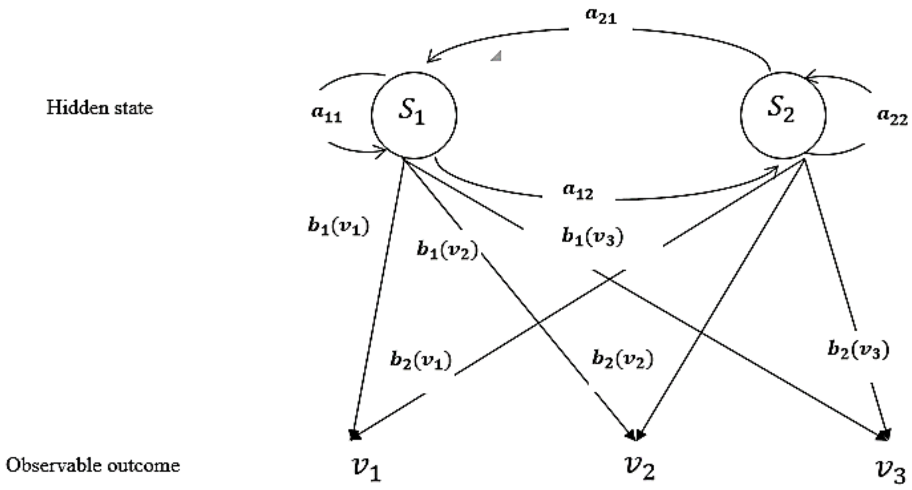

Figure 4.1 Hidden Markov Model with three possible emissions and two hidden states.

4.1.4 Emission Probabilities.

Emission probabilities, also known as observation likelihoods, are a core component of Hidden Markov Models (HMMs). These probabilities define how likely it is for each hidden state to produce a particular observable outcome. In essence, while transition probabilities govern the movement between hidden states, emission probabilities connect the hidden states to the observable data, providing a crucial link between the model's internal states and the observed evidence [75].

Let's consider a Hidden Markov Model with:

- N hidden states $S=\{S_1,S_2,...,S_N\}$,
- M possible observations, $V=\{v_1,v_2,...,v_M\}$.

The emission probabilities are represented by a matrix B where each element $b_j(k)$ within this matrix represents the probability of observing v_k

given the current state is S_j. Formally, the emission probability $b_j(k)$ is defined as:

$b_j(k)=P(v_k \text{ at time } t|q_t=S_j)$

- q_t is the hidden state at time t,
- v_k is the k^{th} observation in the set of possible observations,
- $b_j(k)$ is the probability of emitting observation k when in state j.

For all j and k, emission probabilities must satisfy the following constraints:

1. Non-negativity: $b_j(k) \geq 0$, meaning the probability of any state emitting any observation must be non-negative.
2. Normalization: $\sum_{k=1}^{M} b_j(k) = 1$ for all j, ensuring that the probabilities of emitting any of the possible observations from any given state sum up to 1[76].

This reflects the certainty that the system will emit some observation from its current state. The emission probabilities are often represented in matrix form, with matrix B being an $N \times M$ matrix where N is the number of states and M is the number of possible observations. Emission probabilities allow HMMs to model how observable events or symbols are generated from hidden states [77].

$$B = \begin{pmatrix} b_{11} & b_{12} & \dots & a_{1M} \\ b_{21} & b_{22} & \dots & b_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ b_{N1} & b_{N2} & \dots & b_{NM} \end{pmatrix} \tag{4.2}$$

4.1.5 Initial State Probabilities.

The initial state probabilities in a Hidden Markov Model (HMM) describe the probability distribution over the states at the beginning of the observed sequence, or more formally, the probability of the system being in a particular state when the observation sequence begins. These probabilities are crucial for determining how likely it is that the sequence starts in each possible state [78]. A set of N hidden states $S=\{S_1,S_2,...,S_N\}$, The initial state probabilities are represented by a vector π of length N, where each element π_i represents the probability that the Markov chain will start in state S_i. This is expressed as:

$$\pi=[\pi_1,\pi_2,...,\pi_N] \tag{4.3}$$

where $\pi_i = P(q_1=S_i)$, for $1 \le i \le N$, and q_1 is the state at time t=1.

The initial state probabilities must satisfy the following conditions:

1. Non-negativity: Each probability must be non-negative, i.e., $\pi_i \ge 0$ for all i.
2. Normalization: The probabilities must sum to 1, i.e., $\sum_{i=1}^{N} \pi_i=1$, ensuring that they form a valid probability distribution [79].

4.2 Forward Algorithm:

The Forward Algorithm is a fundamental procedure used in Hidden Markov Models (HMMs) to compute the probability of observing a given sequence of observations. Its primary use is in calculating the likelihood of the observed data given the model parameters [80]. See Appendix 11

4.3 Backward Algorithm.

Backward Algorithm is a procedure used in Hidden Markov Models (HMMs) for calculating the probability of the future observation sequence given the current state. This algorithm plays a crucial role in the computation of model parameters during the training process, especially when used in conjunction with the Forward Algorithm for the Baum-Welch (Expectation-Maximization) training algorithm [81]. See Appendix 12

4.4 Viterbi Algorithm.

The Viterbi Algorithm is a dynamic programming approach used to solve the problem of finding the most likely sequence of hidden states known as the Viterbi path in a context where we deal with Markov processes, such as Hidden Markov Models (HMMs). At its core, the algorithm examines a series of observed events and calculates the sequence of states that is most likely to have resulted in those observations. For example, in a simple weather model, where the states might be "Sunny" or "Rainy", and the observations are "Umbrella" or "No Umbrella", the Viterbi Algorithm helps to predict the sequence of weather conditions based on whether people carry umbrellas. It does this by assigning probabilities to each possible state sequence and choosing the sequence with the highest probability [82].

The Viterbi Algorithm employs a few straightforward equations to accomplish its task. Initially, it calculates the probability of each state being the starting point. As the process moves forward in time, it updates the probability of reaching each state at time t from any state at time $t-1$, taking into account the observation at time t. This update rule can be simplified as: for each state, calculate the product of the maximum probability of arriving from any previous state and the probability of making the current observation from that state. This is repeated for each time step and each state. Finally, backtracking from the last observation, the algorithm identifies the sequence of states (the Viterbi path) with the highest overall probability. This efficient step-by-step calculation ensures that the algorithm can handle even complex models with numerous states and observations in a manageable way [83]. See Appendix 13

4.5 Baum-Welch Algorithm (Expectation-Maximization).

The Baum-Welch Algorithm, a specific instance of the Expectation-Maximization (EM) algorithm, is used for training Hidden Markov Models (HMMs). The goal of the Baum-Welch Algorithm is to find the unknown parameters of an HMM — specifically, the transition probabilities between states, and the emission probabilities of observations from those states. This is particularly useful in situations where we have a sequence of observed data (like speech signals in speech recognition) but do not know the sequence of states that generated them or the exact parameters of the model. The algorithm iteratively refines its estimates of these parameters, aiming to maximize the likelihood of the observed data given the model [84]. See appendix [14]

The Baum-Welch Algorithm operates in two main steps: the Expectation (E) step and the Maximization (M) step. In the E step, the algorithm calculates the expected likelihood of different state sequences given the current estimate of the model's parameters. This involves calculating the probability of transitioning from one state to another and generating the observed symbols from those states. Simplified, this could look like computing the likelihood of "Rainy" and "Sunny" days given observations like "Umbrella" and "No Umbrella," with initial guesses for how likely transitions and observations are. In the M step, the algorithm updates the model's parameters to maximize the likelihood of the observed data. This might involve adjusting the probabilities of transitioning from

"Sunny" to "Rainy" or the likelihood of observing an "Umbrella" when it's "Rainy"[85].

4.6 Application in HMM COVID-19 Predictions.

Constructing a Hidden Markov Model (HMM) involves defining not just the hidden states but also the observation states and their respective probabilities given each hidden state. In the context of COVID-19, if we consider "vaccinated" and "non-vaccinated" as observation states, we can relate these to the hidden states of "Low", "Medium", and "High" (representing levels of new daily cases).

The observation matrix (also known as the emission matrix) contains the probabilities of observing each observable state given each hidden state. Here's a hypothetical matrix based on plausible assumptions:

Table 4.1 The observation matrix

	Vaccinated	Non - Vaccinated
Low	0.8	0.2
Medium	0.5	0.5
High	0.2	0.8

Table 4.1 shows the distribution of vaccinated and non-vaccinated populations across different COVID-19 case states:

Vaccinated (80%): In scenarios with low new COVID-19 cases, there is a high likelihood (80%) of encountering a larger vaccinated population. This underscores the effectiveness of vaccination in curbing virus transmission and reducing new case numbers. Non-vaccinated (20%): Conversely, the probability of encountering non-vaccinated individuals is lower (20%) in low-case scenarios, further emphasizing the efficacy of vaccines in mitigating virus spread.

Vaccinated (50%) and Non-Vaccinated (50%): Here, equal probabilities of vaccinated and non-vaccinated individuals (50% each) indicate an uncertain or intermediate effect of vaccination on new case levels. Possible factors contributing to this scenario include partial vaccination coverage or the emergence of vaccine-resistant virus strains.

Vaccinated (20%): In states with a high number of new cases, the proportion of vaccinated individuals observed is low (20%). This suggests that despite some being vaccinated, the overall impact is insufficient to contain virus spread, possibly due to low vaccine coverage or reduced effectiveness against prevailing strains. Non-Vaccinated (80%): A high probability (80%) of encountering non-vaccinated individuals in high-case scenarios underscores the significant role of vaccination status in virus transmission. Adjustments to this matrix should be made based on actual epidemiological data and expert insights to better reflect accurate probabilities To determine the probability of the hidden states given the observed sequence of vaccinations over 5 days, we can use the Forward Algorithm, which is a standard approach for computing the likelihood of an observation sequence given a Hidden Markov Model (HMM). The sequence you provided is:

- Day 1: Vaccinated
- Day 2: Non-Vaccinated
- Day 3: Vaccinated
- Day 4: Vaccinated
- Day 5: Non-Vaccinated

We need a few more pieces to complete the HMM:

1. The initial probabilities of the hidden states (which we'll assume if not provided).
2. The transition matrix for the hidden states (Low, Medium, High).
3. The observation matrix, which you've suggested in the previous example.

Assume the initial probabilities and use the transition matrix and observation matrix that we discussed earlier. If the initial probabilities weren't specified, a common choice is to distribute them evenly across the states or based on the epidemiological context. Let's define these matrices and calculate the probability of the hidden states given the observation sequence using the Forward Algorithm. The probability of observing the sequence (Vaccinated, Non-Vaccinated, Vaccinated, Vaccinated, Non-Vaccinated) over 5 days, given the Hidden Markov Model setup, is approximately 0.0221 (or 2.21%).

Table 4.2 the probabilities of each hidden state evolved over the 5 days:

	Low	Medium	High
Vaccinated	0.2667	0.1667	0.0667
Non - Vaccinated	0.0494	0.0787	0.0766
Vaccinated	0.0582	0.0252	0.0163
Vaccinated	0.0393	0.0147	0.0043
Non - Vaccinated	0.0063	0.0094	0.0064

Table 4.2 provides forward probabilities for each hidden state over five days, given a sequence of observations. These probabilities indicate the likelihood of the disease being in Low, Medium, or High states each day, based on observed vaccination statuses. The table illustrates how each observation whether vaccinated or non-vaccinated affects the estimated disease state, showcasing significant shifts as new information is acquired daily. Additionally, the final column displays the likelihood of ending in each state on the last day after observing the entire sequence. By summing these final-day probabilities, the total probability of the sequence is calculated. This comprehensive approach facilitates dynamic adjustments to our understanding of the hidden state based on observable actions, such as vaccination status in this context.

4.6.1 Results.

The observation matrix outlines the probabilities of observing "Vaccinated" and "Non-Vaccinated" individuals given the states of Low, Medium, and High: Vaccinated individuals were more likely observed in the Low state (80%) and less likely in the High state (20%). Conversely, non-vaccinated individuals were more likely observed in the High state (80%) and less likely in the Low state (20%).

Regarding sequence probability, the likelihood of observing the sequence "Vaccinated, Non-Vaccinated, Vaccinated, Vaccinated, Non-Vaccinated" for five days, as determined by the Hidden Markov Model, was approximately 2.21%. This statistical insight offers valuable information about the probability of specific sequences occurring within the context of the modelled states and observations.

4.6.2Conclusion.

The Hidden Markov Model highlighted the significant impact of vaccination on the perceived state of COVID-19 cases. The high likelihood of being Vaccinated in Low case scenarios supports the effectiveness of vaccination in controlling the spread. Conversely, a higher probability of Non-Vaccinated in High case scenarios aligns with increased transmission risks among unvaccinated populations. The sequence probability calculation revealed the dynamics of how daily observations of vaccination status influence the estimation of the underlying state of the pandemic, demonstrating the usefulness of HMM in understanding complex processes where direct observation of the state is not possible.

Both models provide valuable insights into the dynamics of COVID-19 cases and the impact of interventions like vaccination. The Markov model emphasizes the state stability and transitions over time, while the HMM provides a nuanced view of how observable outcomes (like vaccination status) can inform us about the underlying state of the disease's spread. These models can be instrumental in planning and evaluating public health strategies.

4.7 Detection of the fake (F) and not fake (T) of the banknotes.

Suppose we have two states of the banknotes fake (F), or not fake (T). The objective is to calculate the probability that it is fake (F), or not fake (T), using the Markov model.

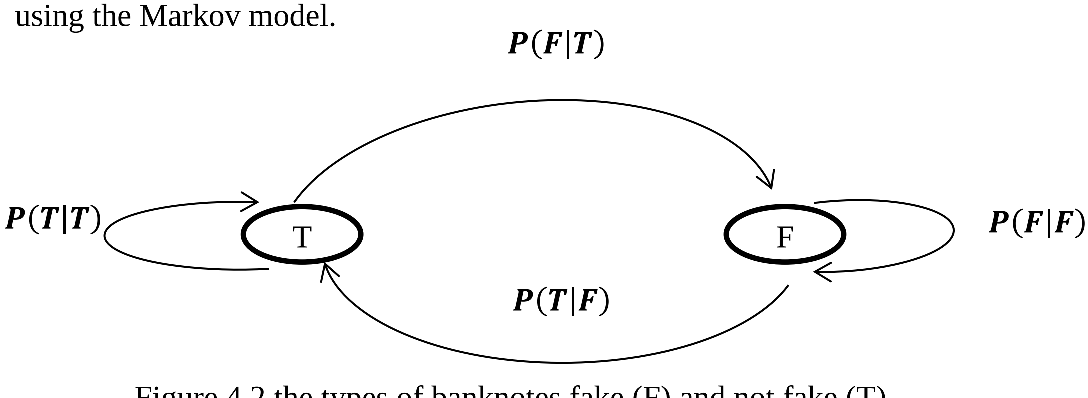

Figure 4.2 the types of banknotes fake (F) and not fake (T)

Fig 4.2 shows the transition states fake (F) and not fake (T) of the banknotes, then the transition matrix

is defined as follows:

$$K=\begin{bmatrix} P(T|T) & P(F|T) \\ P(T|F) & P(F|F) \end{bmatrix}$$

While the initial states matrix is as follows:

$$I = \begin{bmatrix} P(T) \\ P(F) \end{bmatrix}$$

4.8 Detection of the type of banknotes A and B.

Suppose that we have two types of banknotes A and B. The purpose is to know the type of banknotes after depositing them several times.

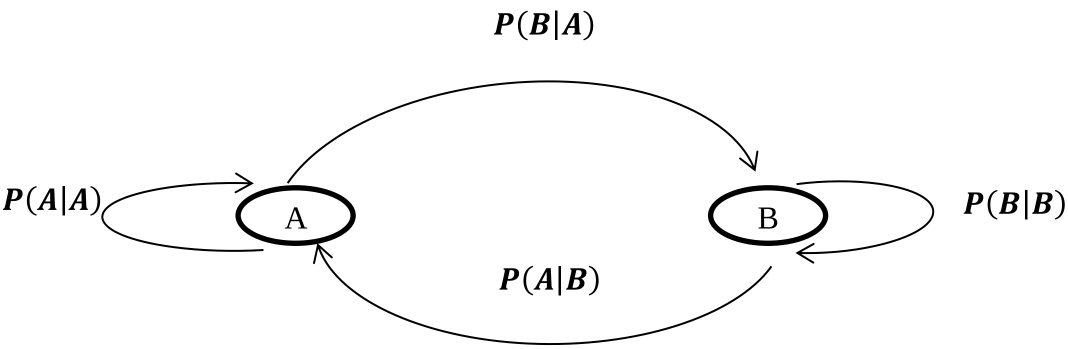

Figure 4.3 Markov model of the values of banknotes A and B

Fig 4.3 shows the transition states of the values A and B of banknotes. Thus, the transition matrix is defined as follows:

$$a_{ij}=\begin{bmatrix} P(A|A) & P(A|B) \\ P(B|A) & P(B|B) \end{bmatrix}$$

The initial states matrix is as follows:

$$\pi = \begin{bmatrix} P(A) \\ P(B) \end{bmatrix}$$

From Bayes theorem

$$\mathrm{P(A|B)} = \frac{\mathrm{P(B|A).P(A)}}{\mathrm{P(B)}}$$

$$\mathrm{P(T|F)} = \frac{\mathrm{P(F|T).P(T)}}{\mathrm{P(F)}}$$

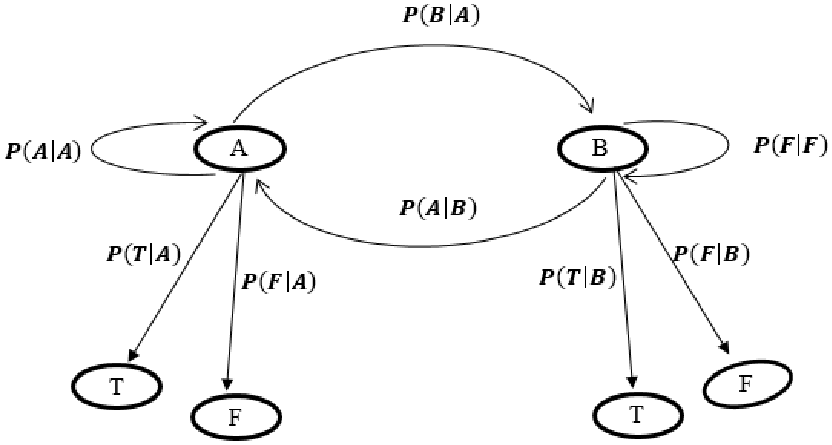

Figure 4.4 Hidden Markov Model for two types of banknotes A and B

In Fig 4.4, we can predict the type of banknotes, whether it is A or B when you know it is fake or not fake, using Bayes' theorem.

Fig 4.5 shows all the different paths of the two processes, A and B, with their different states, T or F, and the path with the highest probability is the most occurring event, which shows that the banknotes is fake or not. Then the probability of all paths in the hidden Markov model are defined as follows:

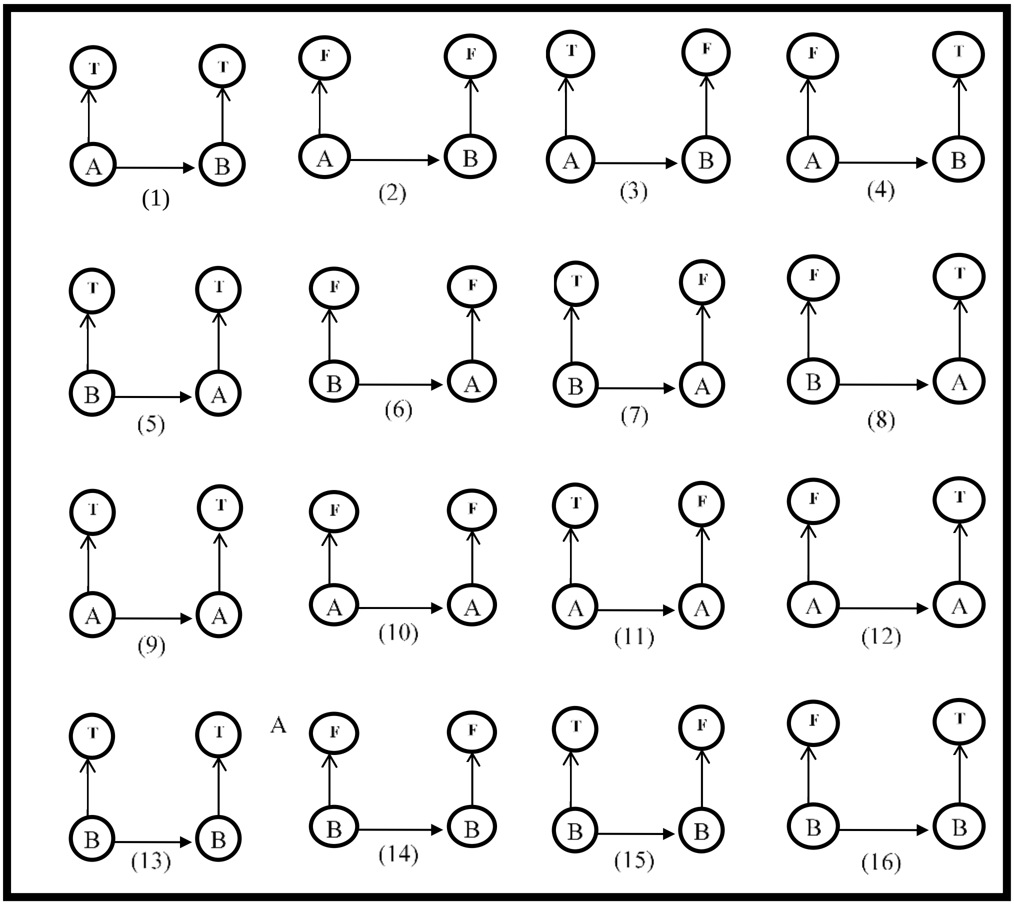

Figure 4.5 All paths of the two values "A" and "B" of the banknotes with their two different states "T" or "F"

4.8.1 Application (1): Predicting the probabilities values of the values A and B of the banknote using the Markov model.

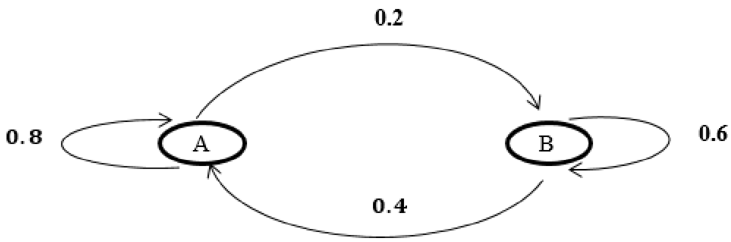

Figure 4.6 Transition states probabilities of the values A and B of the banknotes using Markov model

Fig 4.6 shows the probability values of the transition states of the values A and B of banknotes using the Markov model:

$$a_{ij}=\begin{bmatrix} P(A|A) & P(A|B) \\ P(B|A) & P(B|B) \end{bmatrix} = \begin{bmatrix} 0.8 & 0.4 \\ 0.2 & 0.6 \end{bmatrix}$$

the initial state matrix is

$$\pi_i = \begin{bmatrix} \frac{2}{3} \\ \frac{1}{3} \end{bmatrix}$$

To obtain the probabilities of the cases of the type of currency in the deposit, it is as follows:

$$\text{N} = a_{ij} * \pi_i = \begin{bmatrix} 0.8 & 0.4 \\ 0.2 & 0.6 \end{bmatrix} \begin{bmatrix} \frac{2}{3} \\ \frac{1}{3} \end{bmatrix} = \begin{bmatrix} 0.668 \\ 0.332 \end{bmatrix}$$

Then $P(A) = 0.668$, $P(B) = 0.332$

4.8.2 Application (2): Predicting the probabilities values of the types T and F of banknotes by Markov model

Similarly, for the states of two banknotes fake (F) or not fake (T)

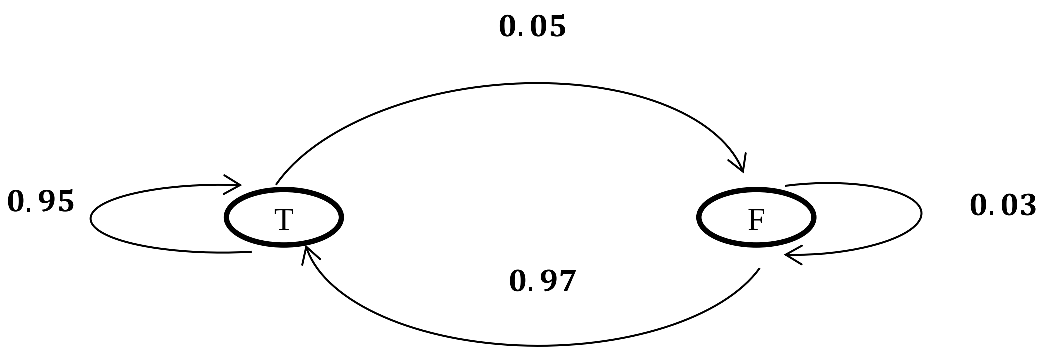

Figure 4.7 Transition states probabilities of the states F and T using the Markov model.

$P(T)$ = 0.99, $P(F)$= 0.01,$P(T|F)$ = 0.97, $P(T|T)$ = 0.95, $P(F|T)$= 0.05, $P(F|F)$= 0.03

Fig 4.7 shows the probability values of the transition states of the types F and T of banknotes using the Markov model.

$$K = \begin{bmatrix} 0.95 & 0.05 \\ 0.97 & 0.03 \end{bmatrix}$$

$$I = \begin{bmatrix} 0.99 \\ 0.01 \end{bmatrix}$$

To obtain the probabilities of transition states of the value of the banknote after a limited number of deposits. Assuming that the number of deposits

is 3.

$$a_{ij}^{(3)} = \begin{bmatrix} 0.8 & 0.4 \\ 0.2 & 0.6 \end{bmatrix} . \begin{bmatrix} 0.8 & 0.4 \\ 0.2 & 0.6 \end{bmatrix} . \begin{bmatrix} 0.8 & 0.4 \\ 0.2 & 0.6 \end{bmatrix} = \begin{bmatrix} 0.688 & 0.624 \\ 0.312 & 0.376 \end{bmatrix}$$

Therefore, the probabilities of moving after three deposits are as follows:

$P(A|A)$ = 0.688

$P(B|B)$ = 0.376

$P(A|B) = 0.312$

$P(B|A) = 0.624$

4.8.3 Application (3): Predicting the probabilities of different paths in HMM

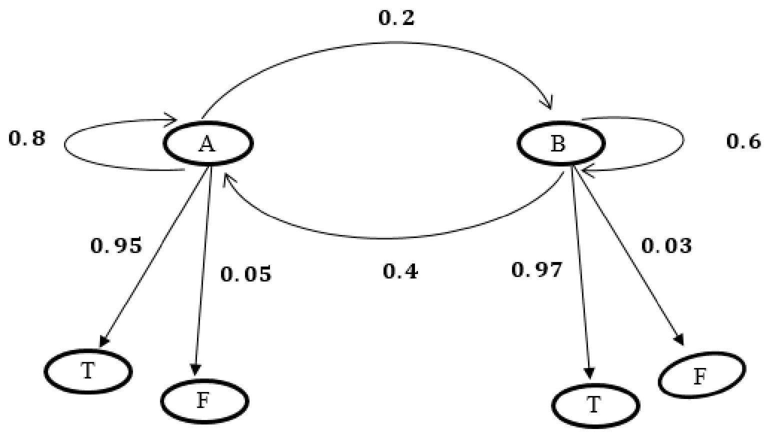

Figure 4.8 Hidden Markov model of the two states of banknotes.

Fig 4.8 represents the hidden Markov model of two banknotes A and B combined with their states Fake (F) or not Fake (T). Where N is the number of hidden states A and B, M represents the states F and T. The symbol $\{a_{ij}\}$ refer to the transition matrix and the symbol $\{b_{jk}\}$ is the observation matrix, while π_i is the initial state matrix.

Table 4.3 The probabilities of different paths for the states of the banknotes, fake or not fake.

i	$Pr(\{path \text{ i}\})$	Probability value
1	$Pr(\{path\ 1\}) = P(A).P(T\|A).P(B\|A).P(T\|B)$	0.123481
2	$Pr(\{path\ 2\}) =P(A).P(F\|A).P(B\|A).P(F\|B)$	0.000201
3	$Pr(\{path\ 3\}) = P(A).P(T\|A).P(B\|A).P(F\|B)$	0.003819
4	$Pr(\{path\ 4\})= P(A).P(F\|A).P(B\|A).P(T\|B)$	0.006499
5	$Pr(\{path\ 5\}) = P(B).P(T\|B).P(A\|B).P(T\|A)$	0.121638
6	$Pr(\{path\ 6\}) = P(B).P(F\|B).P(A\|B).P(F\|A)$	0.000198
7	$Pr(\{path\ 7\}) = P(B).P(T\|B).P(A\|B).P(F\|A)$	0.006402
8	$Pr(\{path\ 8\}) = P(B).P(F\|B).P(A\|B).P(T\|A)$	0.0035739
9	$Pr(\{path\ 9\}) = P(A).P(T\|A).P(A\|A).P(T\|A)$	**0.48374**
10	$Pr(\{path\ 10\})= P(A).P(F\|A).P(A\|A).P(F\|A)$	0.00134
11	$Pr(\{path\ 11\}) = P(A).P(T\|A).P(A\|A).P(F\|A)$	0.02546
12	$Pr(\{path\ 12\}) = P(A).P(F\|A).P(A\|A).P(T\|A)$	0.02546
13	$Pr(\{path\ 13\}) = P(B).P(T\|B).P(B\|B).P(T\|B)$	0.1862982
14	$Pr(\{path\ 14\}) = P(B).P(F\|B).P(B\|B).P(F\|B)$	0.0001782
15	$Pr(\{path\ 15\}) = P(B).P(T\|B).P(B\|B).P(F\|B)$	0.0057618
16	$Pr(\{path\ 16\}) = P(B).P(F\|B).P(B\|B).P(T\|B)$	0.0059499

The data set of all probabilities in HMM were given in Fig (9). Table 4.3 shown the path number (i), its formula and its probability value where i=1, 2,….,16. Consequently, in Fig(10).it was found that the most frequently occurring event is the most probable path 9, when the number of deposits is twice.

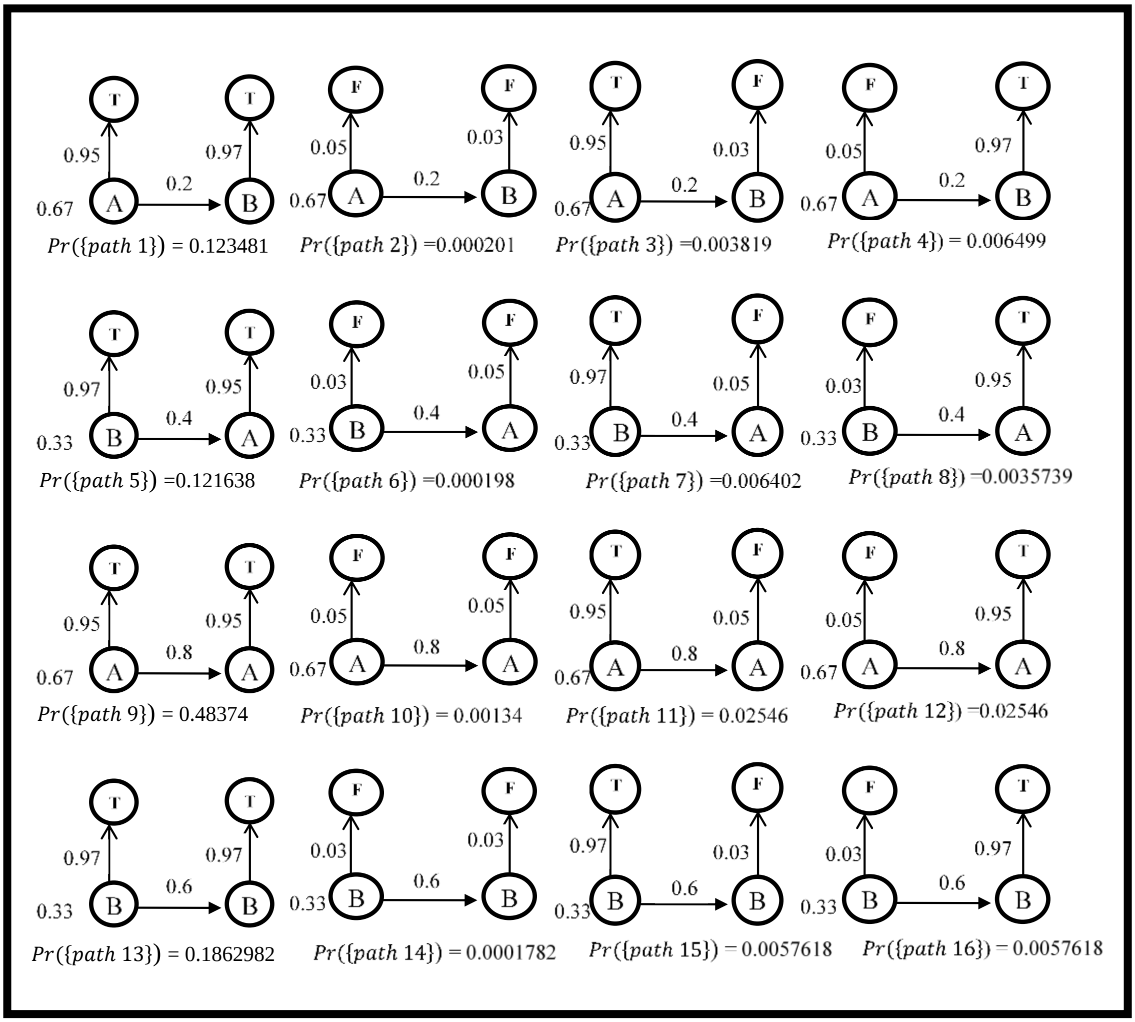

Figure 4.9 All probability values of all paths for the states of banknotes.

4.8.4Conclusion:

This study discovered the probabilities of banknote values and whether if they are fake or not when depositing using the Markov models. In addition, a hidden Markov model was used to determine the chance of observing the trajectory of banknote states on deposit. Researchers can apply this study to other data sets to predict the future data.

Chapter 5

Conclusion and Future Work

Chapter (5) Conclusion and Future Work.

5.1 Conclusion

This thesis has delved into the critical issue of determining banknote values and verifying their authenticity, utilizing advanced statistical models with a primary focus on Markov models and hidden Markov models. The exploration of these models represents a significant contribution to the field of banknote valuation and authenticity assessment, addressing a fundamental challenge in financial security and commerce.

The application of Markov models provided a robust framework for assessing the probabilities associated with banknote values over time. By modeling the transitions between different states of banknote conditions based on observed data, this approach offers a predictive tool that aids in decision-making processes related to currency management and fraud detection. The findings underscore the model's capability to extract meaningful insights from historical data, thereby enhancing the precision and reliability of banknote valuation mechanisms.

Furthermore, the incorporation of hidden Markov models extended the analysis to account for the latent factors influencing banknote states, particularly after repeated circulation. This adaptive modeling approach not only enhances the accuracy of authenticity assessments but also provides a framework for predicting the evolution of banknote conditions under varying environmental and operational scenarios. Such predictive capabilities are crucial for financial institutions and regulatory bodies tasked with maintaining the integrity of currency systems amidst evolving technological and counterfeit threats.

Beyond the domain of currency valuation and authenticity, the thesis has demonstrated the versatility of Markov models in other predictive tasks, such as weather forecasting. The successful application of transition matrix methodologies highlights their potential to uncover temporal patterns and anticipate future conditions based on historical observations. However, it is important to acknowledge the limitations inherent in these models, particularly their sensitivity to contextual factors like seasonal variations and geographic influences, which may impact their predictive accuracy.

Moreover, the utilization of hidden Markov models in analyzing COVID-19 dynamics underscored their utility in public health forecasting. By examining the interplay between vaccination rates and disease spread, the models provided valuable insights into the trajectory of the pandemic and the effectiveness of intervention strategies. This application underscores the broader relevance of statistical models in informing evidence-based decision-making processes across diverse domains, including public health policy and crisis management.

In summary, this thesis contributes substantively to the fields of statistics and applied mathematics by advancing our understanding and application of Markov models and hidden Markov models in complex decision-making scenarios. The insights gained from this research not only enrich theoretical knowledge but also offer practical tools for addressing real-world challenges in currency valuation, fraud detection, weather forecasting, and pandemic response. Moving forward, continued research and refinement of these methodologies hold the promise of further enhancing their predictive power and applicability across a wide range of disciplines and industries.

5.2 Future Work.

Future work in this area could focus on several promising avenues to enhance the predictive capabilities and applicability of the Markov model and Hidden Markov Model (HMM) in weather forecasting and COVID-19 dynamics prediction:

1. **Incorporating Additional Factors**: Expand the Markov model to incorporate additional factors beyond daily case counts, such as demographic information, healthcare capacity, and intervention measures. This could provide a more comprehensive understanding of COVID-19 transmission dynamics and facilitate more accurate predictions of future case trajectories.

2. **Dynamic Parameter Estimation**: Develop methods for dynamically estimating model parameters in both the Markov model and HMM, taking into account changing environmental conditions, seasonal variations, and evolving epidemiological trends. Adaptive algorithms could be employed to continuously update transition and emission probabilities based on incoming data, improving model accuracy and robustness over time.

3. **Spatial and Temporal Analysis**: Conduct spatial and temporal analyses to explore regional variations in both weather patterns and COVID-19 dynamics. This could involve developing spatially explicit models that capture local heterogeneities in transmission dynamics and weather phenomena, allowing for more targeted intervention strategies and forecasting at finer spatial scales.

4. **Integration with Machine Learning**: Explore the integration of machine learning techniques, such as deep learning or ensemble methods, with the Markov model and HMM to improve predictive

accuracy and capture complex nonlinear relationships within the data. This could involve leveraging advanced algorithms to learn intricate patterns from large-scale datasets and enhance the models' predictive capabilities.

5. **Real-time Prediction and Decision Support**: Develop real-time prediction models and decision support systems that integrate weather forecasts, COVID-19 case projections, and intervention strategies. Such systems could provide timely insights to policymakers, healthcare professionals, and meteorologists, enabling proactive planning and response to emerging threats and challenges.

6. **Interdisciplinary Collaboration**: Foster interdisciplinary collaborations between researchers in epidemiology, climatology, computer science, and other relevant fields to leverage domain-specific expertise and methodologies. This could lead to the development of more holistic models that capture the complex interactions between weather patterns, disease dynamics, and societal factors.

7. **Validation and Evaluation**: Conduct rigorous validation and evaluation of the models using independent datasets and metrics to assess their reliability and generalizability. Comparative studies with existing forecasting methods and benchmark datasets can provide insights into the strengths and limitations of the proposed approaches and guide future model development efforts.

By pursuing these avenues for future work, researchers can advance the state-of-the-art in predictive modeling of weather patterns and COVID-19 dynamics, ultimately contributing to more effective planning, mitigation, and response efforts in the face of evolving challenges.

References

References.

[1] Mor, B., Garhwal, S., & Kumar, A. (2021). A systematic review of hidden Markov models and their applications. Archives of computational methods in engineering, 28, 1429-1448.

[2] Stamp, M. (2004). A revealing introduction to hidden Markov models. Department of Computer Science San Jose State University, 26-56.

[3] Adams, S., Beling, P. A., & Cogill, R. (2016). Feature selection for hidden Markov models and hidden semi-Markov models. IEEE Access, 4, 1642-1657.

[4] Chang, L. and Vodka, O., 2024. The HMM-based banknote recognition system. In E3S Web of Conferences (Vol. 522, p. 01053). EDP Sciences.

[5] Kukkonen, J., Olsson, T., Schultz, D. M., Baklanov, A., Klein, T., Miranda, A. I., ... & Eben, K. (2012). A review of operational, regional-scale, chemical weather forecasting models in Europe. Atmospheric Chemistry and Physics, 12(1), 1-87.

[6] Rodríguez, A., Kamarthi, H., Agarwal, P., Ho, J., Patel, M., Sapre, S. & Prakash, B. A. (2022). Data-centric epidemic forecasting: A survey. arXiv preprint arXiv:2207.09370.

[7] Schultz, M. G., Betancourt, C., Gong, B., Kleinert, F., Langguth, M., Leufen, L. H., ... & Stadtler, S. (2021). Can deep learning beat numerical weather prediction?. Philosophical Transactions of the Royal Society A, 379(2194), 20200097.

[8] Marfak, A., Achak, D., Azizi, A., Nejjari, C., Aboudi, K., Saad, E., ... & Youlyouz-Marfak, I. (2020). The hidden Markov chain modelling of the COVID-19 spreading using Moroccan dataset. Data in brief, 32, 106067.

[9] Capobianco, R., Kompella, V., Ault, J., Sharon, G., Jong, S., Fox, S., ... & Stone, P. (2021). Agent-based Markov modeling for improved COVID-19 mitigation policies. Journal of Artificial Intelligence Research, 71, 953-992.
[10] Tian, Y., Luthra, I., & Zhang, X. (2020). Forecasting COVID-19 cases using Machine Learning models. MedRxiv, 2020-07.
[11] Moran, R. J., Fagerholm, E. D., Cullen, M., Daunizeau, J., Richardson, M. P., Williams, S., ... & Friston, K. J. (2020). Estimating required 'lockdown'cycles before immunity to SARS-CoV-2: Model-based analyses of susceptible population sizes,'S0', in seven European countries including the UK and Ireland. MedRxiv, 2020-04.
[12] Li, Z., Wu, Z., He, Y., & Fulei, C. (2005). Hidden Markov model-based fault diagnostics method in speed-up and speed-down process for rotating machinery. Mechanical Systems and Signal Processing, 19(2), 329-339.
[13] Ailliot, P., Allard, D., Monbet, V., & Naveau, P. (2015). Stochastic weather generators: an overview of weather type models. Journal de la société française de statistique, 156(1), 101-113.
[14] Ghosh, S., & Mukherjee, A. (2022). STROVE: Spatial data infrastructure enabled cloud–fog–edge computing framework for combating COVID-19 pandemic. Innovations in Systems and Software Engineering, 1-17.
[15] Huntingford, C., Jeffers, E. S., Bonsall, M. B., Christensen, H. M., Lees, T., & Yang, H. (2019). Machine learning and artificial intelligence to aid climate change research and preparedness. Environmental Research Letters, 14(12), 124007.
[16] Draxl, C., Hodge, B. M., Clifton, A., & McCaa, J. (2015). Overview and meteorological validation of the wind integration national dataset toolkit

(No. NREL/TP-5000-61740). National Renewable Energy Lab.(NREL), Gol den, CO (United States).
[17] Cheval, S., Micu, D., Dumitrescu, A., Irimescu, A., Frighenciu, M., Iojă, C., ... & Antonescu, B. (2020). Meteorological and ancillary data resources for climate research in urban areas. Climate, 8(3), 37.
[18] Ray, P. P. (2016). A survey of IoT cloud platforms. Future Computing and Informatics Journal, 1(1-2), 35-46.
[19] Shuja, J., Alanazi, E., Alasmary, W., & Alashaikh, A. (2021). COVID-19 open source data sets: a comprehensive survey. Applied Intelligence, 51(3), 1 296-1325.
[20] Dong, E., Ratcliff, J., Goyea, T. D., Katz, A., Lau, R., Ng, T. K., ... & Gardner, L. M. (2022). The Johns Hopkins University Center for Systems Science and Engineering COVID-19 Dashboard: data collection process, challenges faced, and lessons learned. The lancet infectious diseases, 22(12), e370-e376.
[21] Zhang, W., Liu, S., Osgood, N., Zhu, H., Qian, Y., & Jia, P. (2023). Using simulation modelling and systems science to help contain COVID-19: A systematic review. Systems research and behavioral science,40(1), 207-234.
[22] Chatfield C. What is the ‘best’method of forecasting?. Journal of Applied Statistics. 1988 Jan 1;15(1):19-38.
[23] Murphy, A.H., 1993. What is a good forecast? An essay on the nature of goodness in weather forecasting. Weather and forecasting, 8(2),pp.281-293.
[24] Zou, K. H., Tuncali, K., & Silverman, S. G. (2003). Correlation and simple linear regression. Radiology, 227(3), 617-628.
[25] Uyanık, G. K., & Güler, N. (2013). A study on multiple linear regression analysis. Procedia-Social and Behavioral Sciences, 106, 234-240.
[26] Schober, P., & Vetter, T. R. (2021). Logistic regression in medical research. Anesthesia & Analgesia, 132(2), 365-366.

[27] Mulenga, E., Bollen, M. H., & Etherden, N. (2021). Distribution networks measured background voltage variations, probability distributions characterization and Solar PV hosting capacity estimations. Electric Power Systems Research, 192, 106979.
[28] Hong, Y. (2013). On computing the distribution function for the Poisson binomial distribution. Computational Statistics & Data Analysis, 59, 41-51.
[29] Laudański, L. M., & Laudański, L. M. (2013). Binomial distribution. Between Certainty and Uncertainty: Statistics and Probability in Five Units with Notes on Historical Origins and Illustrative Numerical Examples, 87-127.
[30] Inouye, D. I., Yang, E., Allen, G. I., & Ravikumar, P. (2017). A review of multivariate distributions for count data derived from the Poisson distribution. Wiley Interdisciplinary Reviews: Computational Statistics, 9(3), e1398.
[31] Jorcin, J. B., Orazem, M. E., Pébère, N., & Tribollet, B. (2006). CPE analysis by local electrochemical impedance spectroscopy. Electrochimica Acta, 51(8-9), 1473-1479.
[32] Fog, A. (2008). Calculation methods for Wallenius' noncentral hypergeo metric distribution. Communications in Statistics—Simulation and Computation®, 37(2), 258-273.
[33] Eriksson, A., Ghysels, E., & Wang, F. (2009). The normal inverse Gauss ian distribution and the pricing of derivatives. Journal of Derivatives, 16(3), 23.
[34] Kundu, D., & Gupta, R. D. (2008). Generalized exponential distribution : Bayesian estimations. Computational Statistics & Data Analysis, 52(4), 1873-1883.
[35] Gómez, Y. M., Bolfarine, H., & Gómez, H. W. (2014). A new extension of the exponential distribution. Revista Colombiana de Estadística, 37(1), 25-34.
[36] Kuipers, L., & Niederreiter, H. (2012). Uniform distribution of sequences. Courier Corporation.

[37] Narkiewicz, W. (2006). Uniform distribution of sequences of integers in residue classes (Vol. 1087). Springer.
[38] Damgaard, C. F., & Irvine, K. M. (2019). Using the beta distribution to analyse plant cover data. Journal of Ecology, 107(6), 2747-2759.
[39] Olkin, I., & Trikalinos, T. A. (2015). Constructions for a bivariate beta distribution. Statistics & Probability Letters, 96, 54-60.
[40] Strickland, J. (2014). Predictive modeling and analytics. Lulu. com.
[41] Lynch, P. (2008). The origins of computer weather prediction and climate modeling. Journal of computational physics, 227(7), 3431-3444.
[42] Wright, J., & Ma, Y. (2022). High-dimensional data analysis with low-dimensional models: Principles, computation, and applications. Cambridge University Press.
[43] Fink, G. A. (2014). Markov models for pattern recognition: from theory to applications. Springer Science & Business Media.
[44] Al-ani, T. (2011). Hidden Markov models in dynamic system modelling and diagnosis. In Hidden Markov models, theory and applications (pp. 27-50).
[45] Rogers, D. P., & Tsirkunov, V. V. (2013). Weather and climate resilience : Effective preparedness through national meteorological and hydrological services. World Bank Publications.
[46] Abdin, A. F., Fang, Y. P., Caunhye, A., Alem, D., Barros, A., & Zio, E. (2023). An optimization model for planning testing and control strategies to limit the spread of a pandemic–The case of COVID-19. European journal of operational research, 304(1), 308-324.
[47] McClintock, B. T., Langrock, R., Gimenez, O., Cam, E., Borchers, D. L. , Glennie, R., & Patterson, T. A. (2020). Uncovering ecological state dynamics with hidden Markov models. Ecology letters, 23(12), 1878-1903.
[48] Zhou, S., Braca, P., Marano, S., Willett, P., Millefiori, L.M., Gaglione,

D. and Pattipati, K.R., 2021. Application of hidden Markov models to analyze, group and visualize spatio-temporal COVID-19 data. IEEE Access, 9, pp.134384-134401.
[49] Reich, N.G., Brooks, L.C., Fox, S.J., Kandula, S., McGowan, C.J., Moore, E., Osthus, D., Ray, E.L., Tushar, A., Yamana, T.K. and Biggerstaff, M., 2019. A collaborative multiyear, multimodel assessment of seasonal influenza forecasting in the United States. Proceedings of the National Academy of Sciences, 116(8), pp.3146-3154.
[50] Hendam, M.M., Mahfouz, K. and Genedi, M., 2024. Prediction Using Markov Model and Hidden Markov Model for the States of Banknotes. Alfarama Journal of Basic & Applied Sciences, 5(2), pp.262-280.
[51] Tolver, A., 2016. An introduction to Markov chains. Department of Mathematical Sciences, University of Copenhagen.
[52] Ghosh, P., Mukhopadhyay, A., Chanda, A., Mondal, P., Akhand, A., Mukherjee, S., Nayak, S.K., Ghosh, S., Mitra, D., Ghosh, T. and Hazra, S., 2017. Application of Cellular automata and Markov-chain model in geospatial environmental modeling-A review. Remote Sensing Applications: Society and Environment, 5, pp.64-77.
[53] Kijima, M., 2013. Markov processes for stochastic modeling. Springer.
[54] Dudin, A. N., Klimenok, V. I., & Vishnevsky, V. M. (2020). The theory of queuing systems with correlated flows (Vol. 430). Cham: Springer.
[55] Grimshaw, S. D., & Alexander, W. P. (2011). Markov chain models for delinquency: Transition matrix estimation and forecasting. Applied Stochastic Models in Business and Industry, 27(3), 267-279.
[56] Ye, F. X., Wang, Y., & Qian, H. (2016). STOCHASTIC DYNAMICS: MARKOV CHAINS AND RANDOM TRANSFORMATIONS. Discrete & Continuous Dynamical Systems-Serie B, (7).

[57] Fu, J. C., & Koutras, M. V. (1994). Distribution theory of runs: a Marko v chain approach. Journal of the American Statistical Association, 89(427), 1050-1058.
[58] Geyer, C. J. (2011). Introduction to Markov chain monte carlo. Handbook of Markov chain monte carlo, 20116022, 45.
[59] Billingsley, P. (1961). Statistical methods in Markov chains. The annals of mathematical statistics, 12-40.
[60] Friston, K. J., Wiese, W., & Hobson, J. A. (2020). Sentience and the origins of consciousness: From Cartesian duality to Markovian monism. Entr opy, 22(5), 516.
[61] Kantz, H., & Schreiber, T. (2004). Nonlinear time series analysis (Vol. 7). Cambridge university press.
[62] Spieler, D. (2014). Numerical analysis of long-run properties for Marko v population models.
[63] Hur, P., Shorter, K. A., Mehta, P. G., & Hsiao-Wecksler, E. T. (2012). Invariant density analysis: Modeling and analysis of the postural control system using markov chains. IEEE Transactions on Biomedical Engineering, 59(4), 1094-1100.
[64] Shmulevich, I., Dougherty, E. R., & Zhang, W. (2002). Gene perturbation and intervention in probabilistic Boolean networks. Bioinformatics, 18(10), 1319-1331
[65] Monthus, C. (2022). Large deviations for metastable states of Markov processes with absorbing states with applications to population models in stable or randomly switching environment. Journal of Statistical Mechanics: Theory and Experiment, 2022(1), 013206.
[66] Meyn, S. P., & Tweedie, R. L. (2012). Markov chains and stochastic stability. Springer Science & Business Media.

[67] Douc, R., Moulines, E., Priouret, P., & Soulier, P. (2018).
Markov chains (Vol. 1). Springer Series in Operations Research and Financial Engineering: Springer International Publishing.

[68] Levin, D. A., & Peres, Y. (2017). Markov chains and mixing times (Vol. 107). American Mathematical Soc.

[69] Awad, M., Khanna, R., Awad, M., & Khanna, R. (2015).
Hidden markov model. Efficient Learning Machines: Theories, Concepts, and Applications for Engineers and System Designers, 81-104.

[70] Mor, B., Garhwal, S., & Kumar, A. (2021). A systematic review of hidden Markov models and their applications. Archives of computational methods in engineering, 28, 1429-1448.

[71] Ruiz-Suarez, S., Leos-Barajas, V., & Morales, J. M. (2022).
Hidden Markov and semi-Markov models when and why are these models useful for classifying states in time series data?. Journal of Agricultural, Biological and Environmental Statistics, 1-25.

[72] Yu, D., Deng, L., Yu, D., & Deng, L. (2015). Hidden Markov models and the variants. Automatic Speech Recognition:
A Deep Learning Approach, 23-54.

[73] Ghassempour, S., Girosi, F., & Maeder, A. (2014).
Clustering multivariate time series using hidden Markov models.
International journal of environmental research and public health, 11(3), 2741-2763.

[74] Maruotti, A. (2011). Mixed hidden markov models for longitudinal data: An overview. International Statistical Review, 79(3), 427-454.

[75] Ariu, D., Tronci, R., & Giacinto, G. (2011). HMMPayl: An intrusion detection system based on Hidden Markov Models. computers & security, 30(4), 221-241.

[76] Helske, S., & Helske, J. (2019). Mixture hidden Markov models for sequence data: The seqHMM package in R. Journal of Statistical Software.
[77] Adams, S., Beling, P. A., & Cogill, R. (2016). Feature selection for hidd en Markov models and hidden semi-Markov models. IEEE Access, 4, 1642-1657.
[78] Langrock, R., & Zucchini, W. (2011). Hidden Markov models with arbitrary state dwell-time distributions. Computational Statistics & Data Analysis, 55(1), 715-724.
[79] Joo, R., Bertrand, S., Tam, J., & Fablet, R. (2013). Hidden Markov models: the best models for forager movements?. PLoS One, 8(8), e71246.
[80] Awad, M., Khanna, R., Awad, M., & Khanna, R. (2015). Hidden markov model. Efficient Learning Machines: Theories, Concepts, and Applications for Engineers and System Designers, 81-104.
[81] Khreich, W., Granger, E., Miri, A., & Sabourin, R. (2010).
On the memory complexity of the forward–backward algorithm. Pattern Recognition Letters, 31(2), 91-99.
[82] Churbanov, A., & Winters-Hilt, S. (2008). Implementing EM and Viterbi algorithms for Hidden Markov Model in linear memory.
BMC bioinformatics, 9, 1-15.
[83] Alhaidari, S., & Zohdy, M. (2019, April). Network anomaly detection using two-dimensional hidden markov model based viterbi algorithm. In 201 9 IEEE International Conference On Artificial Intelligence Testing (AITest) (pp. 17-18). IEEE.
[84] Oudelha, M., & Ainon, R. N. (2010, June). HMM parameters estimation using hybrid Baum-Welch genetic algorithm. In 2010 International Symposium on Information Technology (Vol. 2, pp. 542-545). IEEE.
[85] Ben-Yishai, A., & Burshtein, D. (2004). A discriminative training algorithm for hidden Markov models. IEEE Transactions on Speech and Audio Processing, 12(3), 204-217.

Appendix.

Appendix.

Appendix [1]

Python code to explain and draw Binomial Distribution.

```
import numpy as np

import matplotlib.pyplot as plt

# Parameters for the binomial distribution

n = 10  # number of trials

p = 0.5  # probability of success

# Generate random numbers following binomial distribution

binomial_data = np.random.binomial(n, p, 1000)

# Plot the histogram

plt.hist(binomial_data, bins=np.arange(0, n+2)-0.5, density=True,
alpha=0.75, color='blue', edgecolor='black')

# Plot the theoretical probability mass function

x = np.arange(0, n+1)

pmf = np.array([(np.math.factorial(n) / (np.math.factorial(k) *
np.math.factorial(n - k))) * (p ** k) * ((1 - p) ** (n - k)) for k in x])

plt.plot(x, pmf, 'ro-', linewidth=2)

# Add labels and title

plt.xlabel('Number of Successes')

plt.ylabel('Probability')

plt.title('Binomial Distribution (n={}, p={})'.format(n, p))

# Add legend

please.legend(['Theoretical PMF', 'Simulated Data'])

# Show plot

plt.grid(True)

plt.show()
```

Appendix [2]

Python code to explain and draw Poisson Distribution.

```
import numpy as np

import matplotlib.pyplot as plt

# Parameters for the Poisson distribution

lambda_ = 5  # mean rate of events occurring per unit time or space

# Generate random numbers following Poisson distribution

poisson_data = np.random.poisson(lambda_, 1000)

# Plot the histogram

plt.hist(poisson_data, bins=range(0, max(poisson_data) + 1), density=True,
alpha=0.75, color='blue', edgecolor='black')

# Plot the theoretical probability mass function

x = np.arange(0, max(poisson_data) + 1)

pmf = np.exp(-lambda_) * (lambda_ ** x) / np.math.factorial(x)

plt.plot(x, pmf, 'ro-', linewidth=2)

# Add labels and title

plt.xlabel('Number of Events')

plt.ylabel('Probability')

plt.title('Poisson Distribution (lambda={})'.format(lambda_))

# Add legend

plt.legend(['Theoretical PMF', 'Simulated Data'])

# Show plot

plt.grid(True)

plt.show()
```

Appendix [3]

Python code to explain and draw Geometric Distributions.

```
import numpy as np

import matplotlib.pyplot as plt

# Parameters for the Geometric distribution

p = 0.3  # probability of success in each trial

# Generate random numbers following Geometric distribution

geometric_data = np.random.geometric(p, 1000)

# Plot the histogram

plt.hist(geometric_data, bins=range(1, max(geometric_data) + 2),
density=True, alpha=0.75, color='blue', edgecolor='black')

# Plot the theoretical probability mass function

x = np.arange(1, max(geometric_data) + 1)

pmf = (1 - p) ** (x - 1) * p

plt.plot(x, pmf, 'ro-', linewidth=2)

# Add labels and title

plt.xlabel('Number of Trials until First Success')

plt.ylabel('Probability')

plt.title('Geometric Distribution (p={})'.format(p))

# Add legend

plt.legend(['Theoretical PMF', 'Simulated Data'])

# Show plot

plt.grid(True)

plt.show()
```

Appendix [4]

Python code to explain and draw Hypergeometric Distribution.

```
import numpy as np

import matplotlib.pyplot as plt

from scipy.stats import hypergeom

# Parameters for the Hypergeometric distribution

N = 100  # population size

K = 30   # number of successes in the population

n = 10   # sample size

# Generate random numbers following Hypergeometric distribution

hypergeom_rvs = hypergeom.rvs(N, K, n, size=1000)

# Plot the histogram

plt.hist(hypergeom_rvs, bins=np.arange(0, n+2)-0.5, density=True,
alpha=0.75, color='blue', edgecolor='black')

# Plot the probability mass function (PMF)

x = np.arange(0, n+1)

pmf = hypergeom.pmf(x, N, K, n)

plt.plot(x, pmf, 'ro-', linewidth=2)

# Add labels and title

plt.xlabel('Number of Successes in Sample')

plt.ylabel('Probability')

plt.title('Hypergeometric Distribution (N={}, K={}, n={})'.format(N, K, n))

# Add legend

plt.legend(['Theoretical PMF', 'Simulated Data'])

# Show plot

plt.grid(True)

plt.show()
```

Appendix [5]

Python code to explain and draw Normal (Gaussian) Distribution.

```
import numpy as np

import matplotlib.pyplot as plt

# Parameters for the Normal distribution

mu = 0     # mean

sigma = 1  # standard deviation

# Generate random numbers following Normal distribution

normal_data = np.random.normal(mu, sigma, 1000)

# Plot the histogram

plt.hist(normal_data, bins=30, density=True, alpha=0.75, color='blue',
edgecolor='black')

# Plot the probability density function (PDF)

x = np.linspace(-4, 4, 1000)

pdf = (1/(sigma * np.sqrt(2 * np.pi))) * np.exp(-(x - mu)**2 / (2 *
sigma**2))

plt.plot(x, pdf, 'r-', linewidth=2)

# Add labels and title

plt.xlabel('Value')

plt.ylabel('Probability Density')

plt.title('Normal Distribution (mu={}, sigma={})'.format(mu, sigma))


# Add legend

plt.legend(['Theoretical PDF', 'Simulated Data'])

# Show plot

plt.grid(True)

plt.show()
```

Appendix [6]

Python code to explain and draw Exponential Distribution.

```
import numpy as np

import matplotlib.pyplot as plt

# Parameters for the Exponential distribution

lam = 0.5  # lambda, rate parameter (inverse of mean)

# Generate random numbers following Exponential distribution

exponential_data = np.random.exponential(scale=1/lam, size=1000)

# Plot the histogram

plt.hist(exponential_data, bins=30, density=True, alpha=0.75, color='blue', 
edgecolor='black')

# Plot the probability density function (PDF)

x = np.linspace(0, max(exponential_data), 1000)

pdf = lam * np.exp(-lam * x)

plt.plot(x, pdf, 'r-', linewidth=2)

# Add labels and title

plt.xlabel('Value')

plt.ylabel('Probability Density')

plt.title('Exponential Distribution (lambda={})'.format(lam))

# Add legend

plt.legend(['Theoretical PDF', 'Simulated Data'])

# Show plot

plt.grid(True)

plt.show()
```

Appendix [7]

Python code to explain and draw Uniform Distribution.

```
import numpy as np

import matplotlib.pyplot as plt

# Parameters for the Uniform distribution

a = 0  # lower bound of the distribution

b = 1  # upper bound of the distribution

# Generate random numbers following Uniform distribution

uniform_data = np.random.uniform(a, b, 1000)

# Plot the histogram

plt.hist(uniform_data, bins=30, density=True, alpha=0.75, color='blue',
edgecolor='black')

# Plot the probability density function (PDF)

x = np.linspace(a, b, 1000)

pdf = np.full_like(x, 1/(b-a))

plt.plot(x, pdf, 'r-', linewidth=2)

# Add labels and title

plt.xlabel('Value')

plt.ylabel('Probability Density')

plt.title('Uniform Distribution (a={}, b={})'.format(a, b))

# Add legend

plt.legend(['Theoretical PDF', 'Simulated Data'])

# Show plot

plt.grid(True)

plt.show()
```

Appendix [8]

Python code to explain and draw Beta Distribution.

```
import numpy as np

import matplotlib.pyplot as plt

from scipy.stats import beta

# Parameters for the Beta distribution

alpha = 2   # shape parameter

beta_param = 5  # shape parameter

# Generate random numbers following Beta distribution

beta_data = np.random.beta(alpha, beta_param, 1000)

# Plot the histogram

plt.hist(beta_data, bins=30, density=True, alpha=0.75, color='blue',
edgecolor='black')

# Plot the probability density function (PDF)

x = np.linspace(0, 1, 1000)

pdf = beta.pdf(x, alpha, beta_param)

plt.plot(x, pdf, 'r-', linewidth=2)

# Add labels and title

plt.xlabel('Value')

plt.ylabel('Probability Density')

plt.title('Beta Distribution (alpha={}, beta={})'.format(alpha,
beta_param))

# Add legend

plt.legend(['Theoretical PDF', 'Simulated Data'])

# Show plot

plt.grid(True)

plt.show()
```

Appendix [9]

Python code for example of Markov chains.

```
import numpy as np

# Define the transition matrix

transition matrix = np.array([[0.7, 0.3],

                              [0.4, 0.6]])

# Define the initial state probabilities

initial_state = np.array([0.5, 0.5])

# Function to simulate the Markov chain

def simulate_chain(transition_matrix, initial_state, num_steps):

    current_state = np.random.choice(len(initial_state), p=initial_state)

    states = [current_state]

    for _ in range(num_steps - 1):

        current_state =
np.random.choice(len(transition_matrix[current_state]),
p=transition_matrix[current_state])

        states.append(current_state)

    return states

# Simulate the Markov chain for 10 steps

num_steps = 10

chain_states = simulate_chain(transition_matrix, initial_state, num_steps)

# Print the states visited by the chain

print("States visited by the Markov chain:")

print(chain_states)

States visited by the Markov chain:
[1, 1, 1, 1, 1, 1, 0, 1, 1, 1]
```

Appendix [10]

Python code for example of Ergodic Theorems in Markov Chains.

```
import numpy as np

# Define the transition matrix for the Markov chain

transition_matrix = np.array([[0.7, 0.3],

                              [0.4, 0.6]])

# Function to find the stationary distribution using power iteration method

def stationary_distribution(transition_matrix, num_iterations=1000):

    n = len(transition_matrix)

    pi = np.ones(n) / n  # Initial guess for the stationary distribution

    for _ in range(num_iterations):

        pi = np.dot(pi, transition_matrix)

    return pi

# Find the stationary distribution

pi = stationary_distribution(transition_matrix)

# Print the stationary distribution

print("Stationary Distribution:")

print(pi)

Stationary Distribution:

[0.57142857     0.42857143]
```

Appendix [11]

an example of the Forward Algorithm used in a Hidden Markov Model (HMM)

```
import numpy as np

# Define the HMM parameters

# Transition probabilities (A matrix)

A = np.array([[0.7, 0.3],

              [0.4, 0.6]])

# Emission probabilities (B matrix)

B = np.array([[0.1, 0.4, 0.5],

              [0.6, 0.3, 0.1]])

# Initial state distribution (pi vector)

pi = np.array([0.6, 0.4])

# Observations (sequence of observations)

observations = [0, 1, 2]  # Example observation sequence: [O1, O2, O3]

# Number of states and observations

N = A.shape[0]  # Number of states

T = len(observations)  # Length of the observation sequence

# Initialize the forward probabilities alpha

alpha = np.zeros((N, T))

# Initialization: compute alpha_1(i) for all states i

for i in range(N):

    alpha[i, 0] = pi[i] * B[i, observations[0]]

# Induction: compute alpha_t(i) for t = 2 to T

for t in range(1, T):

    for j in range(N):

        alpha[j, t] = np.sum(alpha[i, t-1] * A[i, j] * B[j,
observations[t]] for i in range(N))

# Termination: compute the probability of the sequence
```

```
P_sequence = np.sum(alpha[i, T-1] for i in range(N))

print("Forward probabilities (alpha):")

print(alpha)

print("\nProbability of the sequence P(O|lambda):")

print(P_sequence)
```

Appendix [12]

An example of the Backward Algorithm used in a Hidden Markov Model (HMM)

```
import numpy as np

# Define the HMM parameters (same as in the Forward Algorithm example)

A = np.array([[0.7, 0.3],

              [0.4, 0.6]])

B = np.array([[0.1, 0.4, 0.5],

              [0.6, 0.3, 0.1]])

pi = np.array([0.6, 0.4])

# Observations (sequence of observations)

observations = [0, 1, 2]  # Example observation sequence: [O1, O2, O3]

# Number of states and observations

N = A.shape[0]  # Number of states

T = len(observations)  # Length of the observation sequence

# Initialize the backward probabilities beta

beta = np.zeros((N, T))

# Initialization: set beta_T(i) = 1 for all states i

for i in range(N):

    beta[i, T-1] = 1

# Induction: compute beta_t(i) for t = T-2 to 1

for t in range(T-2, -1, -1):

    for i in range(N):
```

```
        beta[i, t] = np.sum(beta[j, t+1] * A[i, j] * B[j, observations[t+1]] for j in range(N))

# Termination: compute the probability of the sequence

P_sequence = np.sum(pi[i] * B[i, observations[0]] * beta[i, 0] for i in range(N))

print("Backward probabilities (beta):")

print(beta)

print("\nProbability of the sequence P(O|lambda):")

print(P_sequence)
```

Appendix [13]

Python code for example of Viterbi Algorithm in Hidden Markov Models

```
import numpy as np

# Define the transition matrix A and emission matrix B

A = np.array([[0.7, 0.3],

              [0.4, 0.6]])  # Example transition matrix


B = np.array([[0.1, 0.4, 0.5],

              [0.6, 0.3, 0.1]])  # Example emission matrix

pi = np.array([0.6, 0.4])  # Example initial state distribution

# Define the sequence of observations

observations = [0, 2, 1, 2, 0]

# Viterbi algorithm

def viterbi_algorithm(A, B, pi, observations):

    num_states = A.shape[0]

    T = len(observations

    # Initialize the viterbi matrix and backpointer matrix

    viterbi = np.zeros((num_states, Tbackpointers = np.zeros((num_states, T), dtype=int)

    # Initialize the first column of viterbi matrix using initial state probabilities and emission probabilities

    viterbi[:, 0] = pi * B[:, observations[0]]

    # Iterate over the rest of the observation sequence

    for t in range(1, T):
```

```
        for j in range(num_states):
            # Calculate the maximum probability of reaching state j at time t and store the corresponding backpointer
            max_prob = np.max(viterbi[:, t-1] * A[:, j] * B[j, observations[t]])
            viterbi[j, t] = max_prob
            backpointers[j, t] = np.argmax(viterbi[:, t-1] * A[:, j])
    # Backtrack to find the most likely sequence of states
    best_path_prob = np.max(viterbi[:, -1])
    best_path = [np.argmax(viterbi[:, -1])]
    for t in range(T-2, -1, -1):
        best_path.insert(0, backpointers[best_path[0], t+1])
    return best_path, best_path_prob
# Find the most likely sequence of states and its probability
best_path, best_path_prob = viterbi_algorithm(A, B, pi, observations)

# Print the most likely sequence of states and its probability
print("Most Likely Sequence of States:", best_path)
print("Probability of the Most Likely Sequence:", best_path_prob)
Most Likely Sequence of States: [1, 0, 0, 0, 1]
Probability of the Most Likely Sequence: 0.0008467199999999999
print(A_estimate)
print("\nEstimated Emission Matrix B:")
print(B_estimate)
print("\nEstimated Initial State Distribution pi:")
print(pi_estimate)
Estimated Transition Matrix A:
[[0.48798966 0.51201034]
 [0.51074409 0.48925591]]

Estimated Emission Matrix B:
[[1.04600625e-28 4.63232617e-32 1.00000000e+00]
 [4.99748860e-01 4.99748860e-01 5.02279805e-04]]

Estimated Initial State Distribution pi:
[0. 1.]
```

Appendix [14]

An example of the Baum-Welch algorithm for a simple HMM.

```
import numpy as np

# Example observation sequence

observations = [0, 1, 0, 2, 1]

# Initialize HMM parameters (random initialization)

N = 2  # Number of states

M = 3  # Number of possible observations

A = np.random.rand(N, N)

A /= np.sum(A, axis=1, keepdims=True)  # Normalize rows to sum to 1

B = np.random.rand(N, M)

B /= np.sum(B, axis=1, keepdims=True)  # Normalize rows to sum to 1

pi = np.random.rand(N)

pi /= np.sum(pi)  # Make sure pi sums to 1

# Baum-Welch algorithm

num_iter = 100

epsilon = 1e-6  # Convergence threshold

for iter in range(num_iter):

    # E-step: Calculate alpha, beta, gamma, xi

    alpha, beta, gamma, xi = forward_backward(A, B, pi,
observations)

    # M-step: Update A, B, pi

    # Update transition matrix A

    for i in range(N):

        for j in range(N):
```

```
                    A[i, j] = np.sum(xi[t, i, j] for t in
range(len(observations) - 1)) / np.sum(gamma[t, i] for t in
range(len(observations) - 1))

            # Update emission matrix B

            for j in range(N):

                for k in range(M):

                    B[j, k] = np.sum(gamma[t, j] for t in
range(len(observations)) if observations[t] == k) / np.sum(gamma[t, j] for
t in range(len(observations)))

            # Update initial state distribution pi

            pi = gamma[0]

            # Calculate log-likelihood and check for convergence

            log_likelihood = np.sum(np.log(np.sum(alpha[-1])))

            if iter > 0 and abs(log_likelihood - prev_log_likelihood) <
epsilon:

                break

            prev_log_likelihood = log_likelihood

        # Output final HMM parameters

        print("Final transition matrix A:")

        print(A)

        print("Final emission matrix B:")

        print(B)

        print("Final initial state distribution pi:")

        print(pi)
```

الملخص العربي

تهدف هذه الرسالة إلى تطبيق نموذج ماركوف ونموذج ماركوف المخفي في التنبؤ بقيم الأوراق النقدية وتحديد مدى صحتها، وهو موضوع يحظى باهتمام كبير في مجال التعرف على العملات. تُعدّ مسألة التحقق من صحة العملات وتحديد قيمتها من بين التحديات الرئيسية التي يواجهها الباحثون في مجالات الكشف عن العملات وتقييمها، خاصة مع تزايد الحاجة إلى أنظمة دقيقة وموثوقة للتعرف على العملات في الأنظمة المالية الرقمية.

يعد نموذج ماركوف من الأدوات الفعالة في تقييم الاحتمالات المرتبطة بقيم العملات النقدية، حيث يعتمد على سلسلة من الاحتمالات الانتقالية بين الحالات المختلفة للعملة، مثل كونها سليمة أو تالفة أو مزورة. يستخدم هذا النموذج لفهم وتحليل الأنماط المحتملة التي قد تحدث خلال دورة حياة العملة، سواء من حيث تكرار استخدامها أو تعرضها للتلف.

لكن في حالات أكثر تعقيدًا، قد تكون الحالة الحقيقية للعملة غير مرئية مباشرة، مما يستدعي استخدام نموذج ماركوف المخفي. يتيح هذا النموذج تقدير الحالات الخفية للعملة استنادًا إلى الملاحظات المتاحة مثل عدد الإيداعات أو الاستخدامات المتكررة في النظام المالي. وقد أثبتت الدراسات التجريبية فعالية هذا النموذج في تقديم تنبؤات دقيقة حول الحالة المستقبلية للعملات بعد استخدامها عددًا محددًا من المرات، مما يعزز من دقة أنظمة الكشف ويقلل من الأخطاء المحتملة.

علاوة على ذلك، يمكن تعميم نتائج هذا البحث لتشمل مجالات أخرى تتطلب التنبؤ بالاحتمالات المستقبلية بناءً على بيانات تاريخية، مثل الطقس وانتشار الأوبئة. ففي مجال التنبؤ بالطقس، أظهرت مصفوفات الانتقال التي يعتمد عليها نموذج ماركوف قدرتها على التنبؤ بالحالات الجوية المستقبلية، من خلال ملاحظة الأنماط السابقة مثل ارتفاع درجات الحرارة بعد الأمطار. ومع ذلك، فإن النموذج يواجه بعض القيود، بما في ذلك عدم قدرته على استيعاب التغيرات الموسمية أو الجغرافية. لتجاوز هذه القيود، يأتي دور نموذج ماركوف المخفي، الذي يعتمد على مؤشرات غير مباشرة مثل تكرار استخدام المظلات لتقديم تقديرات أكثر دقة لحالة الطقس

وفي سياق مختلف تمامًا، ساهم استخدام نماذج ماركوف ونماذج ماركوف المخفي في تحليل ديناميكيات انتشار فيروس COVID-19. أظهر نموذج ماركوف المخفي دور التطعيم الكبير في تغيير مسار الوباء وتقليل معدل الانتشار. من خلال دراسة الحالات المسجلة وعدد التطعيمات، قدم هذا النموذج رؤى دقيقة حول كيفية تأثير الأنشطة اليومية على انتشار الفيروس، مما ساعد في تحسين استراتيجيات الصحة العامة.

في الختام، تقدم هذه الرسالة إسهامًا علميًا ملموسًا في استخدام نموذج ماركوف ونموذج ماركوف المخفي لتطبيقات تنبؤية متعددة، بما في ذلك التنبؤ بقيم العملات، الأحوال الجوية، وانتشار الأوبئة. تساعد هذه النتائج في دعم عمليات اتخاذ القرار المستندة إلى البيانات، سواء في المجالات المالية أو الصحية أو البيئية.