

People's Democratic Republic of Algeria  
Ministry of Higher Education for Scientific Research  
University 8 May 45 –Guelma-  
Faculty of Mathematics, Computer Science and Sciences of Matter  
Department of Computer Science



**Master Thesis**

**Specialty:** Computer science

**Option:** Science and Technology of Information and Communication

**Theme**

---

## **An Approach for Handling Missing Data Using Prediction Models**

---

**Presented by :** Kawkab Bouressace

**Jury Members**

**Chairman** Dr. Adel Benamira  
**Supervisor** Dr. Ali Khebizi  
**Examiner** Dr. Aicha Aggoune

**June 2024**

# Acknowledgments

First and foremost, alhamdulillah for making this journey possible. Alhamdulillah, for granting me the opportunity, strength, and capability to undertake this endeavor.

I would like to express my deepest gratitude to my supervisor, Dr. Ali Khebizi, for his invaluable guidance, continuous support, and patience during my graduate studies. His immense knowledge and plentiful experience have been a source of encouragement throughout my academic research.

A special thanks to my family for their love, sacrifices, and unwavering support. Their encouragement and understanding have been my foundation throughout this endeavor. Without their emotional and moral support, this thesis would not have been possible.

I am also grateful to all the faculty members and my colleagues who have provided their support and encouragement during this journey.

Thank you all.

# Abstract

The presence of missing data in datasets poses a major challenge in data analysis, decision-making processes and other activities in various fields that often require specialized methods to deal with them effectively. In this paper, we propose a novel approach to dealing with missing data using models based on machine learning and deep learning, including a hybrid model with statistical and deletion methods. The proposed hybrid model leverages the strengths of Random Forest for structured data and LSTM for time-series data, providing a comprehensive solution for diverse dataset formats with varying proportions of missing data. Experimental results demonstrate the effectiveness of our approach. The hybrid RF\_LSTM model achieves observation accuracy, outperforming Random Forest and LSTM, and through this work, we contribute to solving the problem of missing data by providing an efficient hybrid model that can be largely used in real-world applications.

**Keywords:** Missing data, RF\_LSTM, random forest, LSTM, deletion, statistical, machine learning, deep learning.

# Résumé

La présence de données manquantes dans les ensembles de données pose un défi majeur dans l'analyse de données, les processus de prise de décision et d'autres processus dans divers domaines qui nécessitent souvent des méthodes spécialisées pour les traiter efficacement. Dans ce mémoire de fin d'études, nous proposons une nouvelle approche pour traiter les données manquantes en utilisant des modèles basés sur l'apprentissage automatique et l'apprentissage profond, y compris un modèle hybride avec des méthodes statistiques et de suppression. Le modèle hybride proposé exploite les forces de Random Forest pour les données structurées et de LSTM pour les données de séries chronologiques, offrant une solution complète pour différents formats d'ensemble de données avec des proportions variables de données manquantes. Les résultats expérimentaux démontrent l'efficacité de notre approche. Le modèle hybride RF\_LSTM atteint une précision d'observation, surpassant à la fois Random Forest et LSTM, et à travers ce travail, nous contribuons à la résolution du problème des données manquantes en fournissant un modèle hybride polyvalent et efficace pour les applications du monde réel.

**Mots clés:** Données manquantes, random forest, LSTM, suppression, statistique, apprentissage automatique, apprentissage profond.

# Contents

<b>Acknowledgments</b> . . . . .	<b>i</b>
<b>Abstract</b> . . . . .	<b>ii</b>
<b>Résumé</b> . . . . .	<b>iii</b>
<b>Contents</b> . . . . .	<b>1</b>
<b>List of Figures</b> . . . . .	<b>5</b>
<b>List of Tables</b> . . . . .	<b>7</b>
<b>List of Abbreviations</b> . . . . .	<b>8</b>
<b>General introduction</b> . . . . .	<b>10</b>
<b>I State of the art</b>	<b>12</b>
<b>1 Databases and incomplete data</b> . . . . .	<b>13</b>
1.1 Introduction . . . . .	14
1.2 Data science . . . . .	14
1.3 Conceptual database models . . . . .	15
1.3.1 Hierarchical and network databases . . . . .	15
1.3.2 Relational Databases . . . . .	16
1.3.3 Object-oriented Databases . . . . .	18
1.3.4 NoSQL Databases . . . . .	19
1.4 Physical databases organisation . . . . .	21
1.4.1 Centralized database systems . . . . .	21
1.4.2 Distributed database system . . . . .	22
1.4.3 Cloud database system . . . . .	22
1.5 Missing data issue . . . . .	23
1.6 Reasons for missing data . . . . .	24
1.7 Challenges of missing data . . . . .	25
1.8 Types of missing data . . . . .	25

1.8.1	Missing completely at random (MCAR) . . . . .	25
1.8.2	Missing at random (MAR) . . . . .	26
1.8.3	Missing not at random (MNAR) . . . . .	26
1.8.4	Intermittent missingness . . . . .	26
1.8.5	Systematic missingness . . . . .	26
1.9	Artificial intelligence (AI) . . . . .	27
1.9.1	Machine learning (ML) . . . . .	27
1.9.2	Deep learning (DL) . . . . .	28
1.9.3	Training and testing . . . . .	28
1.10	Conclusion . . . . .	28
<b>2</b>	<b>Existing techniques for handling missing data . . . . .</b>	<b>30</b>
2.1	Introduction . . . . .	31
2.2	Why handling missing data? . . . . .	31
2.3	How missing data are handled ? . . . . .	32
2.4	Deletion techniques . . . . .	33
2.4.1	Listwise-deletion . . . . .	33
2.4.2	Pairwise-deletion . . . . .	34
2.4.3	Entire variables-deletion . . . . .	34
2.5	Missing data imputation . . . . .	35
2.5.1	Statistical techniques . . . . .	35
2.5.2	Predictive models . . . . .	38
2.6	Conclusion . . . . .	48
<b>3</b>	<b>Related work . . . . .</b>	<b>49</b>
3.1	Introduction . . . . .	50
3.2	Presentation of the research methodology . . . . .	50
3.3	Selection process and article acquisition overview . . . . .	51
3.4	State of the art analysis . . . . .	52
3.4.1	Related works analysis . . . . .	52
3.4.2	Synthesis of related works . . . . .	56
3.5	Conclusion . . . . .	56
<b>II</b>	<b>Modeling and implementation of the approach . . . . .</b>	<b>58</b>
<b>4</b>	<b>The hybrid approach for handling missing data . . . . .</b>	<b>59</b>
4.1	Introduction . . . . .	60
4.2	Framework features . . . . .	60
4.3	Areas of framework usage . . . . .	61
4.4	Architecture of the proposed framework . . . . .	62
4.4.1	Components of the architecture . . . . .	63

---

---

## CONTENTS

---

4.4.2	Interaction between the system components . . . . .	64
4.5	Input / output of the conceived system . . . . .	64
4.6	Dataset description . . . . .	66
4.7	System functionalities . . . . .	66
4.7.1	Storing and merging datasets . . . . .	66
4.7.2	Detection and identification of missing data . . . . .	67
4.7.3	Input dataset statistics . . . . .	68
4.7.4	Missing data imputation . . . . .	68
4.7.5	Display calculated data . . . . .	75
4.7.6	Statistics of the output dataset . . . . .	75
4.7.7	Handling a permanent dataset . . . . .	75
4.8	Usage scenario . . . . .	75
4.9	Conclusion . . . . .	78
<b>5</b>	<b>Implementation and experiments of the solution . . . . .</b>	<b>79</b>
5.1	Introduction . . . . .	80
5.2	Hardware environmental development . . . . .	80
5.3	The software environment . . . . .	80
5.3.1	PyCharm . . . . .	80
5.3.2	Python language . . . . .	81
5.3.3	Python libraries . . . . .	82
5.4	System overview . . . . .	83
5.4.1	Interfaces . . . . .	83
5.4.2	Main modules . . . . .	85
5.5	Exploration of H2MD functionalities . . . . .	86
5.5.1	Handling missing data in various file formats . . . . .	86
5.5.2	Handling missing data with easyPHP . . . . .	90
5.6	Results and analysis . . . . .	93
5.6.1	Results of statistical methods . . . . .	94
5.6.2	Results of deletion methods . . . . .	95
5.6.3	Results of artificial intelligence methods . . . . .	96
5.7	Discussion . . . . .	101
5.8	Conclusion . . . . .	103
	<b>General conclusion . . . . .</b>	<b>104</b>
	<b>Bibliography . . . . .</b>	<b>106</b>
	<b>Annex</b>	
	<b>Startup Creation . . . . .</b>	<b>111</b>
	1.1 Project presentation . . . . .	111

---

---

## CONTENTS

---

1.1.1	The project idea (the proposed solution)	111
1.1.2	The suggested values	112
1.1.3	The working team	112
1.1.4	Project development and deployment plan	112
1.1.5	The project timeline	113
1.2	Innovative aspects	114
1.2.1	The nature of innovations	114
1.3	Areas of innovation	114
1.3.1	New Processes	114
1.3.2	Enhanced functionalities	114
1.3.3	New clients	115
1.3.4	New offers	115
1.3.5	New models	115
1.4	Strategic market analysis	115
1.4.1	Market segment	115
1.4.2	Measuring competition intensity	116
1.4.3	Marketing strategy	116
1.5	Financial plan	117
1.5.1	Costs and charges	117
1.5.2	Revenue forecast	117

---



# List of Figures

1.1	Structure of hierarchical Databases (Database Management System (DBMS)) [1]	15
1.2	Example of hierarchical databases	16
1.3	Example of network databases [2]	16
1.4	Example of Object-oriented Databases [3]	18
1.5	Example of mongoDB [4]	19
1.6	Example of key-value ( <i>redis</i> ) [3]	20
1.7	Example of column-oriented ( <i>Apache Cassandra</i> ) [5]	20
1.8	Example of graph database with Neo4j [6]	21
1.9	Architecture of a centralized database system [7]	22
1.10	Architecture of a distributed database system [8]	22
1.11	Cloud database system [9]	23
2.1	Techniques for dealing with missing data	32
2.2	Example of mean imputation (MI) [10]	36
2.3	Example of median imputation [11]	37
2.4	Example of mode imputation [12]	38
2.5	Simple linear regression plot [13]	39
2.6	Artificial intelligence methods for missing data imputation	42
2.7	ANN model mechanism [14]	43
2.8	Missing data imputation with RNN [15]	44
2.9	Imputing missing sales data with RNN	44
2.10	Imputing missing temperature data with LSTM	45
2.11	Predicting budget data using a decision tree [16]	47
2.12	Example of random forest [17]	48
3.1	Methodology model	50
4.1	Architecture of the proposed approach	62
4.2	The various input formats	65
4.3	Framework outputs	65
4.4	Example of storing	67
4.5	Example of detection and identification of missing data	67
4.6	Example of input dataset statistics	68

---

## LIST OF FIGURES

---

4.7	Example of suggested methods for handling missing data . . . . .	69
4.8	An example of the operation of statistical techniques . . . . .	70
4.9	An example of the operation of deletion techniques . . . . .	71
4.10	Hybrid (RF-LSTM) model mechanism . . . . .	74
4.11	Example of missing data imputation . . . . .	74
4.12	Example of display calculated data . . . . .	75
4.13	Number of missing values in each column . . . . .	76
4.14	Selecting predictive model for imputation . . . . .	77
5.1	Pycharm community 2022-03 . . . . .	81
5.2	Python logo [18] . . . . .	82
5.3	Splash screen . . . . .	84
5.4	Home interface . . . . .	84
5.5	English main interface . . . . .	84
5.6	Arabic main interface . . . . .	85
5.7	The main modules of H2MD . . . . .	85
5.8	Example of uploading various datasets formats . . . . .	86
5.9	Illustration of the method ( RF_LSTM ) and its features . . . . .	87
5.10	Example of processing missing data using RF_LSTM . . . . .	87
5.11	Statistics illustration of handling missing data . . . . .	88
5.12	Example of processing data using Listwise deletion . . . . .	88
5.13	Comparison of the used methods . . . . .	89
5.14	Illustration of downloading the processed dataset in csv format . . . . .	90
5.15	Example of amanaging a dataset in easyPHP format . . . . .	90
5.16	Example of how to select a database from a table or dataset . . . . .	91
5.17	Illustration of selecting the pairwise method and features specification . . . . .	91
5.18	Example of selected features for data analysis . . . . .	92
5.19	Example of a processed dataset using RF_LSTM . . . . .	92
5.20	Graph of statistical methods across different levels of missing data . . . . .	95
5.21	Graph of deletion methods across different levels of missing data . . . . .	96
5.22	Graph of random forest model across different levels of missing data . . . . .	97
5.23	Training loss . . . . .	97
5.24	Graph of LSTM across different levels of missing Data . . . . .	98
5.25	Training loss over epochs . . . . .	99
5.26	Graph of RF_LSTM across different levels of missing data . . . . .	100
5.27	Graph comparing other models with RF_LSTM Across Different Levels of Missing Data . . . . .	100

---

# List of Tables

1.1	Example of a relational database table for products . . . . .	17
1.2	Student dataset example . . . . .	23
2.1	Example about listwise-deletion [19] . . . . .	33
2.2	Example about pairwise-deletion[19] . . . . .	34
2.3	Example about entire variables-deletion [19] . . . . .	35
2.4	Dataset: years of experience, education level, age, and salary before imputation . . . . .	40
2.5	Dataset: years of experience, education level, age, and salary After imputation . . . . .	40
2.6	Daily temperature readings with missing values . . . . .	45
2.7	Predicting missing data using KNN . . . . .	46
3.1	Google scholar results for missing data related keyword phrases since 2021	52
3.2	Summary of existing work focusing on missing data . . . . .	55
4.1	Description of the system components . . . . .	63
4.2	An excerpt of london weather dataset . . . . .	66
4.3	Incomplete dataset CSV . . . . .	76
4.4	Complete dataset by mode method . . . . .	77
4.5	Dataset imputation with predictive models . . . . .	78
5.1	Characteristics of the used hardware . . . . .	80
5.2	Statistical methods across different levels of missing data . . . . .	94
5.3	Comparison of methods . . . . .	95
5.4	Random forest across different levels of missing data . . . . .	96
5.5	LSTM across different levels of missing data . . . . .	98
5.6	RF_LSTM across different levels of missing data . . . . .	99
5.7	Comparison of research study techniques using different approaches . . . .	102
1.1	Project tasks timeline: (Mo) Month . . . . .	113
1.2	Subscription data: one year (N) . . . . .	117

# List of abbreviations

**AI** Artificial Intelligence

**JSON** Binary JavaScript Object Notation

**CRM** Customer Relationship Management

**CSV** Comma-Separated Values

**DB** Databases

**DBMS** Database Management System

**DL** Deep Learning

**EDI** Electronic Data Interchange

**GPA** Grade Point Average

**H2MD** Hybrid Handling Missing Data

**IDE** Integrated Development Environment

**IoT** Internet of Things

**JSON** JavaScript Object Notation

**MAR** Missing at Random

**MCAR** Missing Completely at Random

**ML** Machine Learning

**MNAR** Missing Not at Random

**NoSQL** Not Only Structured Query Language

**PDF** Portable Document Format

**RDB** Relational Databases

**RDBMS** Relational Database Management Systems

**SQL** Structured Query Language

**XML** EXtensible Markup Language

**XLSX** Microsoft Excel Open EXtensible Markup Language (XML) Spreadsheet

# General introduction

Today, the world of business and management is oriented towards decision-making, which is fundamentally based on an efficient analysis of data managed by organizations. In the context of globalization, where competition is increasing, the decision-making process requires high-quality data. A fundamental characteristic of data quality is the absence of missing data, which can arise for various reasons. This complete data forms the basis on which predictions are made and correlations are operated. However, in reality, such complete and flawless data is rare, as missing data, which can arise for many reasons, such as human error or malfunction in data collection processes, poses various challenges that can significantly impact the work of data analysts and interpreters. Till then, missing data introduces uncertainty, bias, and complexity into statistical analyses, reducing the accuracy of conclusions and biasing results. Thus, handling missing data requires the selection of distinct techniques such as statistical methods, deletion methods, or artificial intelligence-based. Thus, handling.

Taking into account that each dataset has its own characteristics that can appear in different forms, such as the number of features and the type of data used for each feature, the type of data set itself, whether general or time series, in addition to the specific format used when storing the file containing data in permanent storage supports, such as a CSV file, creating a system robust enough to handle this missing data is a major challenge. This system needs to be adaptable to different characteristics of data sets, and to achieve this, extensive research must be conducted and the best solutions found to achieve the best results. In this end-of-study report, we propose a comprehensive system for dealing with diverse missing data through various methods, along with a hybrid method that enables us to deal with datasets. Our work is organized into two parts: the first part is dedicated to the state of the art and contains three chapters, the second part includes two chapters, focusing on key aspects of solution modeling, system development and experimental results, and their discussion.

**Chapter One:** This chapter is dedicated to exploring database paradigm and the realm of incomplete data. We provide a comprehensive overview of database types and the nature of missing data, including its various causes, types, and missing consequences.

**Chapter Two:** This chapter is dedicated to exploring various techniques for processing missing data. We conduct a comprehensive review of current methods, including an in-depth examination of artificial intelligence-based approaches.

**Chapter Three:** This chapter delves into the existing literature on methods and techniques for handling missing data. It explores foundational studies dealing with missing data, review articles, and academic papers discussing various methodologies.

**Chapter Four:** This chapter focuses on the modeling of our framework, we delve into all stages that constitute this framework.

**Chapter Five:** The fifth chapter addresses the implementation phase of the system development project and presents the work environment and the results obtained from evaluating algorithm performance.

# **Part I**

## **State of the art**



Chapter **1**

## Databases and incomplete data

## 1.1 Introduction

Databases (DB) are characterized as organized stores for organizing, putting away, and recovering colossal sums of data in different shapes and sorts. However, in the real world, these specifications can suffer from incomplete or missing data, which presents challenges when dealing with them. Missing data can arise as a result of many factors. Incomplete data is critical because it can affect the accuracy and reliability of analyses and decision-making processes.

This first chapter examines the various forms of DB, in addition to identifying incomplete data, the risks that they can cause, and the factors that appear in them, in addition to the types of missing data. We start by presenting the overall domain of data science, which includes DB and data analytics.

As the two concepts, missing data and incomplete data, refer to data that is expected but is absent, these two concepts are used interchangeably in this report. We start by presenting the general field of data science.

## 1.2 Data science

Data science is an interdisciplinary area that analyzes large DB using scientific methods, algorithms, and procedures. In this perspective, data scientists leverage a combination of skills in computer science, statistics, and business to extract insights from data. The main goal is to discover useful information, informing conclusions, and support decision-making.

Different technologies are deployed in data science, and data scientists utilize various technologies in their work, including [20]:

- Artificial Intelligence (AI): They employ machine learning models and associated software to predict outcomes.
- Cloud Computing: Data scientists leverage cloud technologies to gain the flexibility and data processing capacity needed for advanced analytics.
- Internet of Things (IoT): Data science projects benefit from data collected by interconnected IoT devices, which are accessible over the internet.

Within data science projects, datasets commonly contain missing values, and identifying and addressing missing data is crucial, as many statistical methods rely on complete datasets as input. Failing to handle missing data appropriately can lead to inaccurate predictive models or algorithm failure, making it an essential phase in any data science project [21, 22]. Furthermore, missing data may impact decision support closely.

As data collection can be structured in different ways, in the following section, existing models for organizing data are exposed.

## 1.3 Conceptual database models

There are different data models for specifying data and their relationships, each of which results in a corresponding database type. These types, designed to handle various data needs, are presented below:

### 1.3.1 Hierarchical and network databases

Hierarchical and network **DB** indeed belong to earlier generations of **DBMS** and are considered outdated compared to modern relational and Not Only Structured Query Language (**NoSQL**) Databases. Here's a brief overview of each type:

#### A) Hierarchical databases

This database type organizes data in a parent-child relationship, creating nodes structured like a tree. This data is stored in records linked together, where each child record is linked to only one parent, while each parent record can have multiple child records [23].

Figure 1.1 depicts the hierarchical structure of a **DBMS**, organized in a tree-like structure.

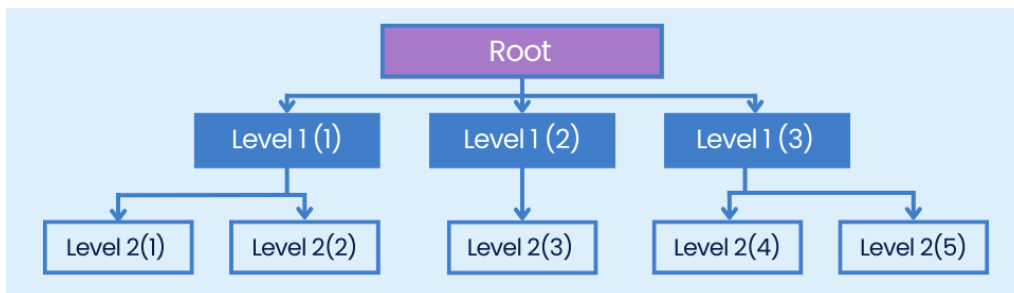


Figure 1.1: Structure of hierarchical Databases (**DBMS**) [1]

**Example 1.** *In a hierarchical database, each child record has only one parent record. For instance, in a banking database model, the "custaccu" record serves as the parent for both the "account" and "customer" records. Each child record represents specific attributes, so the "account" record may include details like "number" and "position" while the "customer" record could have information such as "street", "city" and "name" (see figure 1.2).*

*Hierarchical databases rely on this clear "parent-child" structure for well-organized data management.*

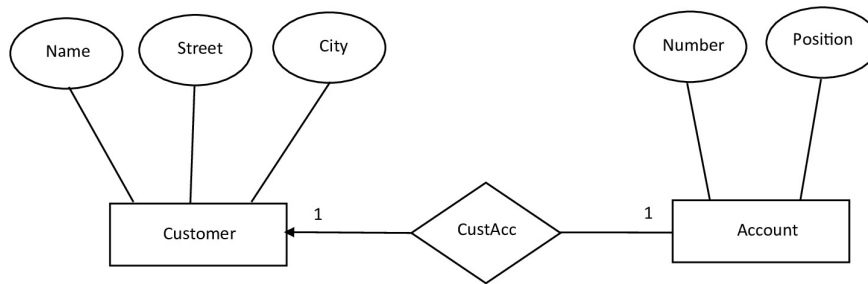


Figure 1.2: Example of hierarchical databases

### B) Network databases

Generally, this kind of database follows the network data model where data is represented in this model as nodes connected by links. Further, it allows a record to have more than one child, unlike hierarchical [24].

**Example 2.** For example, in the network database of students (see figure 1.3). As we can see, the subject entity has a relationship with both the student entity and the degree entity. So there is an edge connecting the subject entity with both the student and the degree. The subject entity has two parents, and the other two entities have one child entity.

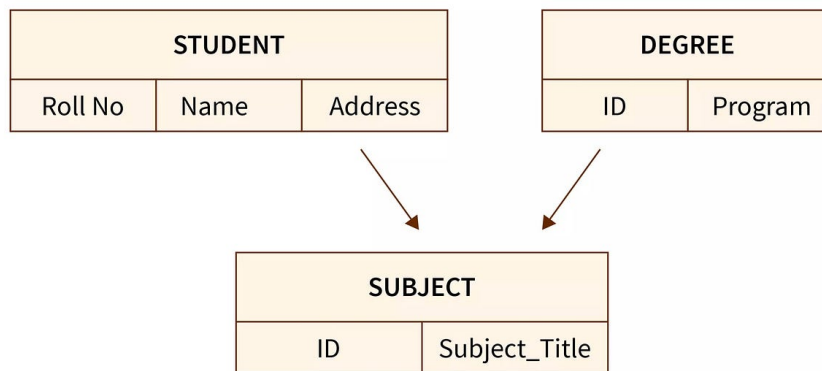


Figure 1.3: Example of network databases [2]

### 1.3.2 Relational Databases

A relational database is a type of database that stores, manipulates, and maintains data in uniform, organized tables. E.F. Codd invented the database in 1970. Every table within the database contains a unique key that distinguishes its data from that of others, and this has become a disadvantage for Relational Databases (RDB) due to the rapid development of applications [25, 26]. To manage relational databases, a software engine called a database management system DBMS is needed.

Examples of relational **DBMS** include MySQL, Microsoft Structured Query Language (**SQL**) Server, Oracle, and others.

By steadfastly following the four essential **ACID** properties - atomicity, consistency, isolation, and durability - we ensure the utmost integrity of our database [25].

- **Atomicity:** Transactions must be treated as indivisible units, ensuring they either succeed entirely or fail completely.
- **Consistency:** Data must always be maintained in a valid state, preserving integrity.
- **Isolation:** Concurrent transactions must yield results consistent with a sequential execution order.
- **Durability:** Transactions committed successfully must persist even in the event of system failures, ensuring data remains intact.

### Challenges of relational Databases

While widely used and highly effective in many scenarios, also come with a set of challenges. Some of the key challenges include [27]:

- Mismatch between object-oriented and relational paradigms.
- Limited adaptability of the relational data model across diverse domains.
- Complex schema evolution due to rigid data models.
- Suboptimal distributed availability stemming from scalability limitations.
- Performance degradation caused by joins, **ACID** transactions, and stringent consistency requirements, particularly in distributed setups.

**Example 3.** *We can take a relational database related to products as an example. In this structure, each row represents a unique product, while each column represents a specific attribute of that product, such as product ID, name, description, price, and category. This setup enables efficient organization and retrieval of product-related information, streamlining tasks like inventory management, sales tracking, and product analysis (see table 1.1).*

Table 1.1: Example of a relational database table for products

Product ID	Name	Description	Price	Category
001	Laptop	High-performance laptop	999 DA	Electronics
002	Smartphone	5G-capable smartphone	799 DA	Electronics
003	T-shirt	Cotton T-shirt	20 DA	Clothing
004	Headphones	Noise-canceling headphones	149 DA	Electronics
005	Watch	Waterproof sports watch	199 DA	Accessories

### 1.3.3 Object-oriented Databases

To store data inside the database system, an object-oriented database uses the object-based data model, similar to objects in object-oriented programming languages, data is represented and saved as objects [28, 23].

**Example 4.** *An example of an object-oriented database system in the context of transport could be a system designed to manage a fleet of vehicles (see figure 1.4).*

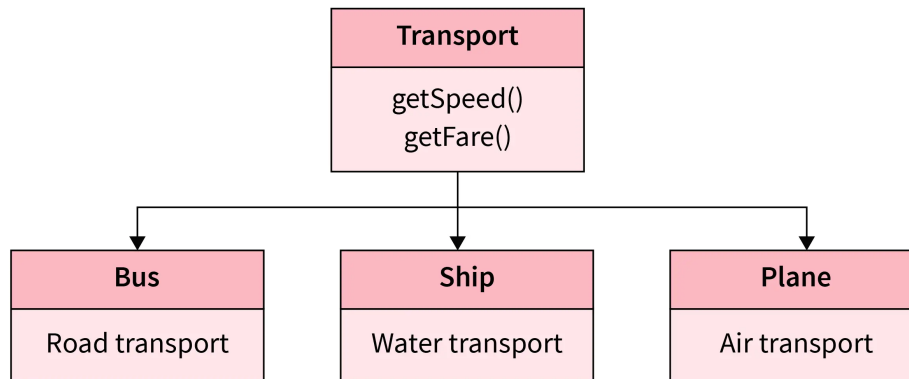


Figure 1.4: Example of Object-oriented Databases [3]

*In this context Transport, Bus, Ship and Plane are classified as objects.*

- *Bus is associated with Road Transport as its characteristic.*
- *Ship is characterized by Water Transport.*
- *Plane is characterized by Air Transport.*

*Transport serves as the fundamental object, from which Bus, Ship and Plane inherit their attributes [3].*

### 1.3.4 NoSQL Databases

The trouble of the inefficiency of relational DB with massive amounts of records has led to the emergence of NoSQL Databases as a powerful opportunity [29].

NoSQL diverges from the traditional Relational Database Management Systems (RDBMS), NoSQL operates as a non-relational database, it stands out as an uncomplicated yet robust solution for managing extensive datasets, enabling seamless scaling and replication on a global scale without reliance on a master configuration, and this database excels in accommodating diverse data types without the need for predefined relationships [30, 25, 31]. It is observed that many kinds of NoSQL Databases have been used both in the literature and in industrial areas. In what follows, we will expose the most commonly used ones, and each type of database will be illustrated with an example.

#### A) Document type database

These NoSQL Databases are called document DB and are powerful at handling data that is kind of structured. They store and find data in document formats like JavaScript Object Notation (JSON), XML, or Binary JavaScript Object Notation (BSON) effectively.

**Example 5.** *One example of a document database is MongoDB. It saves information in documents that look like JSON, making it easy to change data design and handle complex structures. Many people pick MongoDB because it's simple, can grow, and can manage complex data (see figure 1.5).*

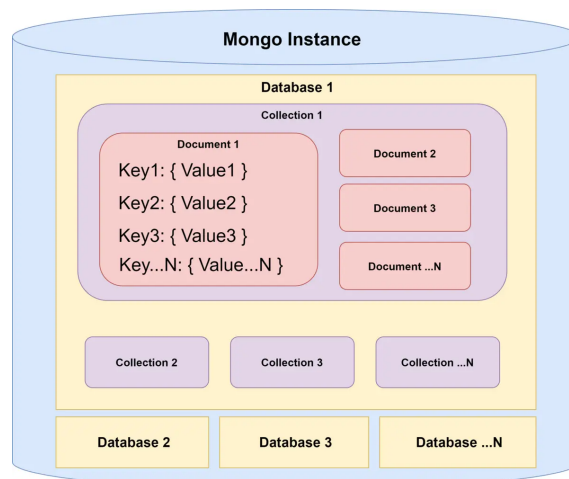


Figure 1.5: Example of mongoDB [4]

#### B) Key-value type database

This specific category, known as key-value stores works on a concept; storing values in a map structure where each value is linked to a unique key. One for the name and one for the value enables high-performance execution of operations making them

top performers in terms of speed. They are commonly utilized for caching, session storage, and managing metadata.

**Example 6.** *Redis is an illustration of a key-value store database. It enables users to store data in a format where each piece of information is linked to a key (see figure 1.6).*

A Document	Key	Value
<pre>{   "BookID": "978-1449396091",   "Title": "Redis - The Definitive Guide",   "BookID": "Salvatore Sanfilippo",   "Year": "2021", }</pre>	BookID	978-1449396091
	Title	Redis - The Definitive Guide
	Author	Salvatore Sanfilippo
	Year	2021

Figure 1.6: Example of key-value (*redis*) [3]

### C) Column-oriented type database

Column-oriented data storages represent a departure from relational DB. Unlike RDBMS, which store data in rows and read row via row, column-oriented DB, as the name implies, store information in columns, the variety of columns can vary from one file to some other, which avoids locating columns with zero values.

**Example 7.** *Apache Cassandra is what is called a column-focused database. Each column in Cassandra signifies a specific feature or sector. The setup lets it store and grab data smoothly, especially when analysis needs data from multiple columns added together (see figure 1.7).*

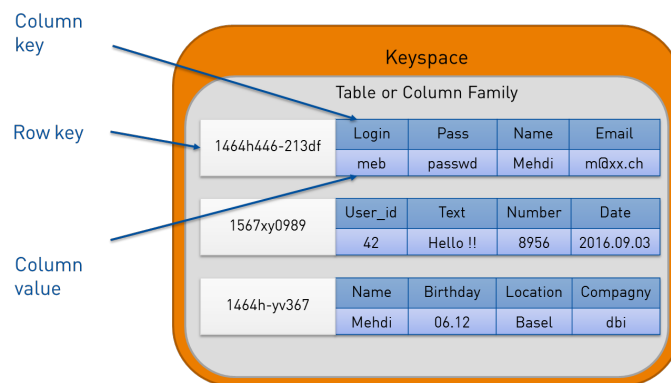


Figure 1.7: Example of column-oriented (*Apache Cassandra*) [5]

### D) Graph type database

A graph database is an oriented graph composed of a set of nodes that represent entities such as people, objects, or concepts, linked with edges that represent the relationships or connections between these entities. These kinds of systems are



perfect for working with connected data quickly. But they might not work as well with data that's more spread out.

**Example 8.** *The best example that can be taken of this type is Neo4j, which is a graph database that represents data as graphs. It is based on the notion of nodes, relationships, and properties. It is well-suited for scenarios like social networks (where in social networks, each user is represented as a node, and his friendships or connections are represented as relationships between nodes), recommendation engines (see figure 1.8).*

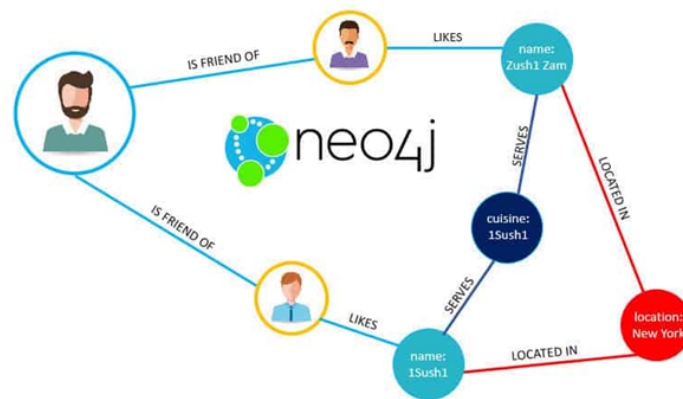


Figure 1.8: Example of graph database with Neo4j [6]

## 1.4 Physical databases organisation

In another perspective, the previous database models are stored on physical storage mediums, necessitating a description of the architecture at the physical level. The most popular systems for storing data are outlined below:

### 1.4.1 Centralized database systems

In a centralized database system, both the **DBMS** and the database itself are housed at a single location, which serves as the central point of access for multiple other systems (see figure 1.9). This centralized setup facilitates streamlined data management and ensures uniform access and control over the database across various interconnected systems [32].

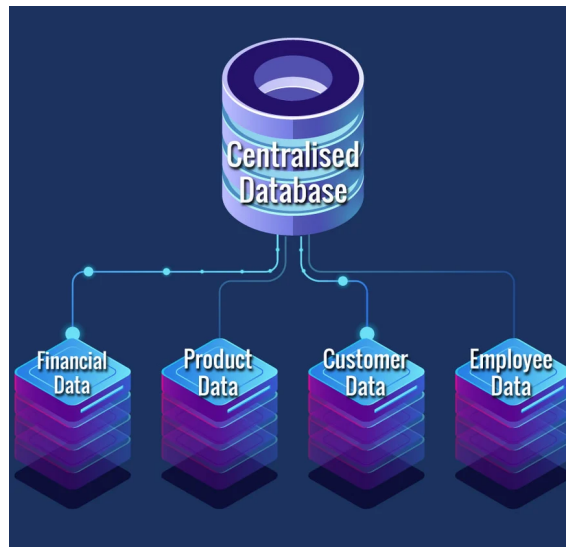


Figure 1.9: Architecture of a centralized database system [7]

### 1.4.2 Distributed database system

In a distributed database system, the physical database and the **DBMS** software are spread across numerous locations connected by a computer network (*see figure 1.10*). Each site typically hosts a portion of the database, ensuring data availability and enabling efficient retrieval and manipulation across the network [32].

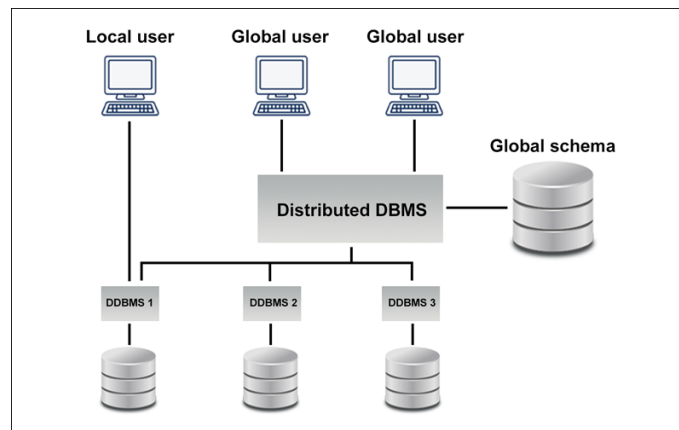


Figure 1.10: Architecture of a distributed database system [8]

### 1.4.3 Cloud database system

Cloud databases function within a cloud infrastructure, where data is stored across off-site servers in private, public, or hybrid cloud environments accessible via the internet. They offer straightforward installation and reconfiguration, facilitating teams in swiftly assessing, authenticating, and experimenting with new business ideas [33]. Figure 1.11 depicts a cloud database system.



Figure 1.11: Cloud database system [9]

After having exposed the necessary background useful for understanding data, their models, and organization, we now tackle one of the most related issues of data, which is missing data.

## 1.5 Missing data issue

Missing data refers to information that is not fully represented or recorded within the database. This incompleteness leads to a lack of comprehensive information for certain aspects of the dataset and can stem from various factors such as missing values. Such incompleteness can significantly impact decision-making and analysis processes, and so on . . . [34, 35].

To illustrate this concept of missing data, we provide the following example.

**Example 9.** We use a dataset containing student information as an example of missing data, which includes age, gender, Grade Point Average (*GPA*), and attendance records. However, due to various reasons, certain data points are missing (see table 1.2):

Table 1.2: Student dataset example

StudentID	Age	Gender	GPA	Attendance
1	18	M	3.5	Yes
2	17	F	NULL	Yes
3	NULL	F	3.9	No
4	16	M	3.2	Yes
5	18	NULL	3.8	Yes
6	17	F	3.7	No

As observed in the previous table, in some student datasets, there are instances where information may be missing, represented as 'NULL' (Not a Number) values. The records containing missing data are:

- *Student 2 is missing the [GPA](#).*
- *Student 3 is missing the age.*
- *Student 5 is missing the gender.*

*This absence of data can create challenges in analyzing academic performance trends and so on.*

Missing data within datasets can manifest for a multitude of reasons. Understanding these underlying factors and effectively managing the resulting challenges are pivotal for maintaining the integrity and dependability of data analysis and interpretation. In what follows these reasons are presented.

## 1.6 Reasons for missing data

Missing data in datasets can be caused by a wide range of factors. These elements consist of the following [30]:

- **Data Entry Errors:** Sometimes, data may not have been recorded properly or may have been lost during entry.
- **Non Response:** Respondents might decline to respond to a question due to concerns about privacy or because they find the question unclear or difficult to understand.
- **Data Processing Errors:** Errors can occur during data processing, manipulation, or transfer, leading to missing values.
- **Data privacy concerns:** There are times when certain information gets excluded from a dataset in order to protect people's privacy or confidentiality.
- **Natural Causes:** Sometimes, data might be missing due to natural causes, such as technical failures or environmental factors affecting data collection.
- **Data Not Relevant:** In some cases, certain variables may not be applicable or relevant to all cases in the dataset, resulting in missing values for those cases.
- **Data Loss:** Data can be lost due to storage or transfer errors, hardware malfunctions, or other technical issues.
- **Incomplete Data Collection:** Data collection efforts may have been incomplete, either due to limited resources, time constraints, or logistical challenges.
- **Intentional Omissions:** In certain situations, data may be intentionally omitted or withheld for various reasons, such as to manipulate results or hide information.
- **Legal or Regulatory Compliance:** In order to comply with legal or regulatory requirements, some sensitive or private data may need to be removed from the

dataset.

- **Withdrawal of Participants:** Data gaps may result from participants quitting surveys or longitudinal studies at any point or from not providing any data at all.

## 1.7 Challenges of missing data

Ignoring missing data can contribute to many risks that pose a real challenge, including the following [35, 36]:

- Inaccurate data can produce biased findings, which can distort the analysis and possibly lead to false conclusions.
- When important data points are absent, decision-makers may encounter difficulties developing strategies or policies that work, which could lead to less-than-ideal results.
- Lack of complete information about the needs or preferences of the target population may cause decision-makers to allocate resources inefficiently, wasting money or missing opportunities.
- It is challenging to draw accurate conclusions in the absence of complete data, which raises the possibility of making claims that are unsupported or inaccurate.
- Uncertainties surrounding missing data can cause stakeholders' faith in decision outcomes to erode, eroding their confidence in the decision-making process.
- The validity of analyses may be jeopardized by incomplete datasets' inability to generate statistically significant relationships or trends.

## 1.8 Types of missing data

The nature of missing data can vary, and understanding the different types is essential for implementing appropriate handling techniques, which include the following types:

### 1.8.1 Missing completely at random (MCAR)

The data is said to be Missing Completely at Random (**MCAR**) when the occurrence of missing values in the dataset is purely by chance. The observed values in the dataset represent a random sample from the complete dataset, had there been no missing values. For instance, this can happen when respondents inadvertently skip questions [28].

**Example 10.** *In a study examining the link between sleep duration and cognitive performance, missing data on sleep duration due to participants accidentally skipping a questionnaire page represents a case of **MCAR**.*

### 1.8.2 Missing at random (MAR)

In this scenario, the missing data can be forecasted based on the other variables in the study rather than being predictable solely from the missing data themselves [34].

**Example 11.** *In a health survey, women are less likely to report their weight than men, but this tendency can be explained by the observed variable "gender".*

### 1.8.3 Missing not at random (MNAR)

When data are Missing Not at Random (MNAR), the absence of data is systematically linked to unobserved variables. In other words, the missingness is associated with events or factors that the researcher did not measure or account for [34].

**Example 12.** *In a study examining income levels and spending habits, participants may refuse to disclose their income if they earn high salaries, leading to MNAR data. This missingness is systematically related to the unobserved variable of high income, potentially biasing the results if not properly accounted for.*

### 1.8.4 Intermittent missingness

Intermittent missingness refers to sporadic instances where data is missing or unavailable at irregular intervals within a dataset, Intermittent missingness can occur unpredictably and sporadically throughout the dataset [37, 38].

**Example 13.** *The weather monitoring system gathers information about temperature, moisture, and wind speed, but sometimes the sensors that do this job can have short-term problems because of the weather or technical issues. This can lead to gaps in the data collected, which can disrupt its flow and make it harder to make sense of.*

### 1.8.5 Systematic missingness

Systematic missingness occurs when there is a pattern or reason behind the missing data, which is not entirely random, unlike MCAR or Missing at Random (MAR), systematic missingness implies that the missing data is related to some underlying factor or mechanism [39, 40].

**Example 14.** *For example, in a medical study tracking patients' adherence to a treatment regimen, systematic missingness might arise if patients with certain health conditions are less likely to attend follow-up appointments due to transportation barriers. This pattern of missing data is not random and is influenced by the patients' health status, creating systematic missingness.*

As we will use artificial intelligence techniques in the development of our approach, we present the basic concepts of this field below and we direct readers to specialized references for further in-depth study of the the artificial intelligence domain.

## 1.9 Artificial intelligence (AI)

Artificial intelligence (AI) refers to the simulation of human intelligence processes by machines, notably computer systems. These processes include learning, reasoning, and self correction. AI applications encompass a large range of tasks, from problem solving to decision making.

The goal of artificial intelligence is to enable machines to perform tasks requiring human intelligence, achieved through machine learning and deep learning techniques [41].

### 1.9.1 Machine learning (ML)

Machine learning refers to algorithms enabling computers to learn from data autonomously. It includes supervised learning, where models are trained on labeled data, unsupervised learning, which identifies patterns in unlabeled data and reinforcement learning, which optimizes actions through rewards [42]. ML finds application in diverse domains, like:

- Recommendation systems are an essential application of machine learning. These systems leverage algorithms to analyze user preferences and behavior, recommending items or content tailored to individual tastes and interests.
- Image recognition, a vital domain within machine learning and computer vision, focuses on developing algorithms capable of identifying and interpreting objects, patterns, and scenes within images or videos.

In machine learning, several algorithms are widely used across various tasks and domains. Some notable ones include:

- **Linear regression:** Used for predicting a continuous variable based on one or more input features.
- **Decision trees:** Employed for both classification and regression tasks, utilizing a tree like model of decisions.
- **Support vector machines (SVM):** Used for classification and regression tasks, particularly in high-dimensional spaces.
- **K-nearest neighbors (KNN):** A non-parametric method for classification and regression and making predictions based on the majority vote of the k nearest data points.

### 1.9.2 Deep learning (DL)

In the realm of artificial intelligence, Deep Learning (DL) stands out as a powerful subset and revolutionizing how machines learn from data through intricate neural networks, these networks consist of interconnected layers of neurons that process information hierarchically and enabling them to learn complex patterns, representations directly from raw data [43]. DL finds application in diverse domains like:

- **Computer vision:** DL powers advanced image recognition, object detection, segmentation, and image generation tasks. Computer Vision: DL powers advanced image recognition, object detection, segmentation, and image generation tasks.
- **Natural language processing (NLP):** DL models excel in language understanding, sentiment analysis, machine translation, text generation and question answering systems.

In DL, several algorithms are widely used across various tasks and domains. Some notable ones include:

- **Convolutional neural networks (CNNs):** Especially effective for image recognition tasks due to their ability to extract hierarchical features.
- **Recurrent neural networks (RNNs):** Ideal for sequential data processing such as natural language processing and time series analysis.

### 1.9.3 Training and testing

In the context of artificial intelligence, the process of training and testing serves as a critical phase for evaluating the efficacy and performance of AI models.

- **Training:** the model is exposed to labeled data, allowing it to learn patterns and relationships within the dataset through optimization algorithms.
- **Testing:** involves evaluating the trained model's performance on unseen data to assess its generalization capabilities and potential for real-world deployment.

Key metrics such as accuracy, precision, recall, and F1 score are commonly used to quantify the model's effectiveness in making predictions or classifications.

## 1.10 Conclusion

In this first chapter, the basic concepts related to data are exposed, as well as different database models and their respective storage technologies. We then delved into the various types of missing data, and their fundamental causes were deeply explained, as well as the potential challenges they may pose. The question that now arises is: How best to address



these missing data problems, and what are the existing techniques used to face the related challenges?

The forthcoming chapter tries to answer the previous questions.

# Chapter 2

## Existing techniques for handling missing data

## 2.1 Introduction

Handling missing data is a major challenge for the data analysis process. In fact, incomplete datasets, which can be a big issue, are tackled by multiple techniques that aim to face missing data issues. These methods include data imputation approaches and deletion techniques, the exploitation of domain knowledge, and more, such techniques are used to resolve incomplete data and to confirm the reliability of analyses. In what follows, we emphasize two important questions related to missing data. The first one is **why handling data is a very important challenge to be tackled in the field of data science**, and the second question focuses on **how these techniques operate**.

## 2.2 Why handling missing data?

Dealing with missing data often involves addressing the following aspects that are commonly encountered [44]:

- **Preservation of data integrity:** By handling missing data, we ensure that the dataset remains accurate and reliable.

**Example 1.** *In a medical study analyzing patient outcomes, handling missing medical records ensures that the dataset accurately reflects the health status of all patients, preserving the integrity of the research findings.*

- **Validity of the analysis:** Proper handling of missing data ensures that the analysis is based on complete and faithful information, leading to valid conclusions and informed decision-making.

**Example 2.** *In a market research survey, addressing missing responses ensures that the analysis accurately represents consumer opinions, leading to valid market trend predictions.*

- **Precision of the results:** Handling missing data appropriately can improve the precision of the results by reducing uncertainty and variability in the dataset.

**Example 3.** *In climate modeling, properly accounting for missing weather data such as temperature readings at certain locations increases the precision of regional climate projections.*

- **Risk reduction:** Effective handling of missing data helps mitigate the risks associated with incomplete or unreliable information, minimize the potential for erroneous conclusions, inaccurate predictions, and suboptimal decisions.

**Example 4.** *In financial analysis, addressing missing financial data values reduces the risk of making erroneous investment decisions based on incomplete or unreliable financial information, thus minimizing potential financial losses.*

- **Enhanced research reproducibility:** Handling missing data in a systematic and transparent manner improves the reproducibility of research findings, allowing other researchers to replicate analyses and verify results effectively.

**Example 5.** *In social science studies, transparently documenting and handling missing survey responses increases the reproducibility of research findings.*

### 2.3 How missing data are handled ?

The exploration of the research literature shows that two fundamental approaches are used to resolve missing data issues. As depicted in figure 2.1, the first one consists of deleting rows or columns, while the second one focuses on imputation. The following section addresses the question relating to the interest attached to the handling of missing data.

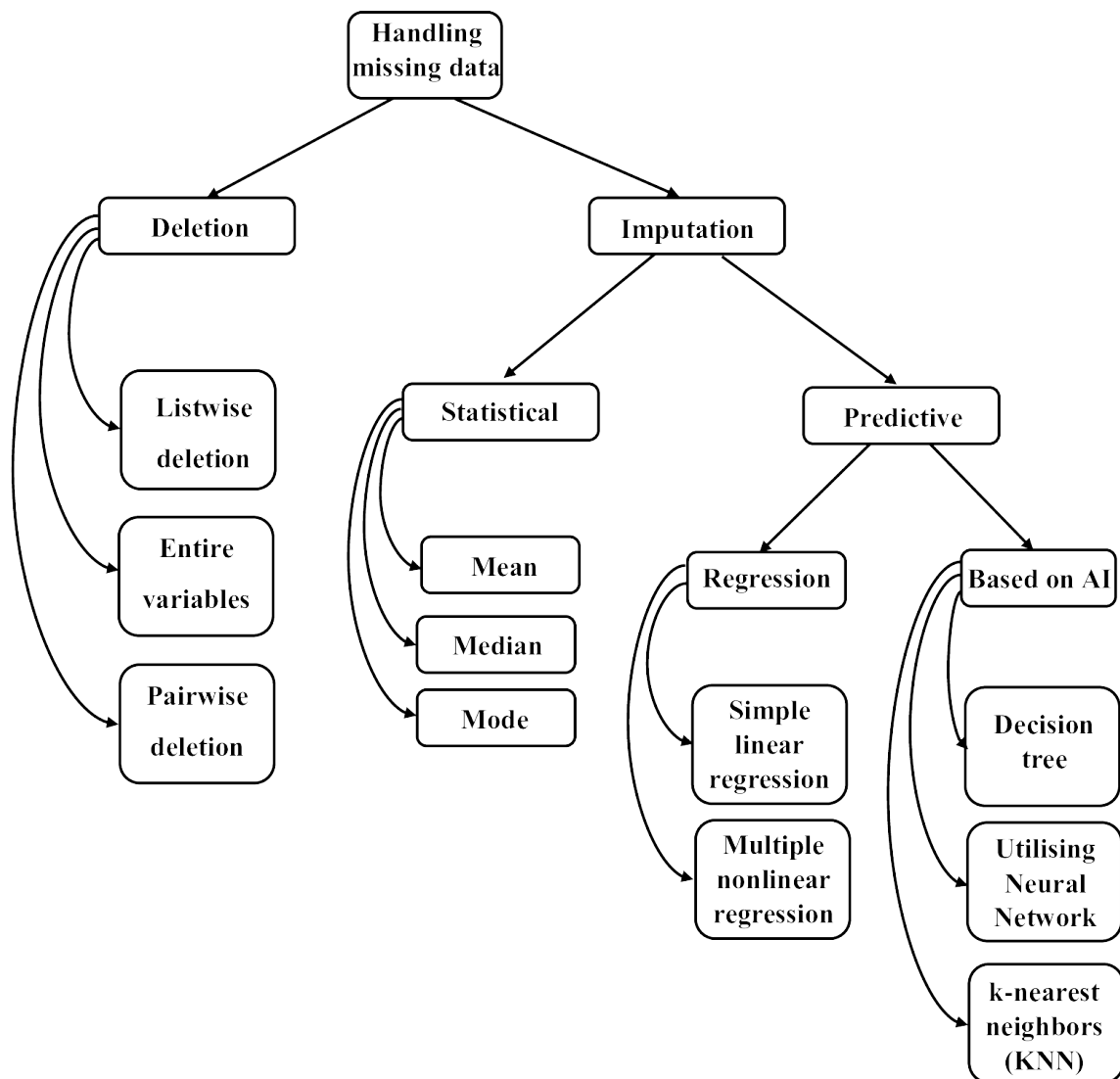


Figure 2.1: Techniques for dealing with missing data

## 2.4 Deletion techniques

For handling missing data, deletion techniques consist simply to remove records or variables with no values from the dataset in hand. There are three types of deletion techniques:

### 2.4.1 Listwise-deletion

The listwise deletion method or the complete-case analysis procedure involves the removal of observations from the dataset in case there are any missing values (*i.e.*, *NULL*) contained in those observations. This will result in a missing values free dataset so that the analysis algorithms could be applied to the final dataset that has been generated from the smaller dataset that was initially missing values free (*see table 2.1*) [45].

Table 2.1: Example about listwise-deletion [19]

ID	Gender	Depression Rating	Favorite Color
1	Male	0	Blue
2	Male	2	Green
3	Female	1	Red
4	Male	4	NULL
5	Female	5	Yellow
6	Female	9	Purple
7	Male	3	Green
8	Female	4	Blue
9	Female	NULL	Blue
10	Male	8	Red

In the provided example of table 2.1, applying listwise deletion involves removing any rows that have missing values in any of the specified columns. In this case, rows 4 and 9 have missing values in the Favorite Color and Depression Rating columns, respectively. Row 4 has a NULL value in the Favorite Color column, indicating that the information about the Favorite Color for the corresponding individual is missing. Similarly, row 9 has a NULL value in the Depression Rating column, indicating missing information about the individual's Depression Rating.

By applying listwise deletion, we would remove these rows from the dataset to ensure that our analysis is based only on complete cases where all relevant information is available.

While it may seem practical to remove cases with missing data, especially if they are few in number, this method often proves disadvantageous. Listwise deletion relies on the assumption of *MCAR*, which is rarely met in practice. Hence, it can create biased results if listwise deletion is employed. Additionally, it can lead to a great loss of data in cases where there are huge numbers of absent values. Indeed, if many observations contain missing values, listwise deletion will remove those observations entirely and result in a

fairly small dataset. In such cases, subsequent analyses may have some inaccuracies or low reliability.

### 2.4.2 Pairwise-deletion

Pairwise deletion's a method for handling missing data in which only the missing values for specific variables are ignored when performing calculations for those variables. This means that any analysis involving a pair of variables will use all available data for that specific pair, even if one or both variables have missing values for other observations (*see figure 2.2*) [46].

Table 2.2: Example about pairwise-deletion[19]

ID	Gender	Depression Rating	Favorite Color
1	Male	6	Blue
2	Male	2	Green
3	Female	1	Red
4	Male	4	NULL
5	Female	5	Yellow
6	Female	9	Purple
7	Male	3	Green
8	Female	4	Blue
9	Female	NULL	Blue
10	Male	8	Red

In the example of table 2.2, we find pairwise deletion for Depression Rating and Favorite Color:

When employing pairwise deletion for the analysis of Depression Rating, only the rows with missing values in this specific column are excluded. For instance, in this dataset, row 9 with a NULL Depression Rating would be omitted from calculations specifically pertaining to Depression Rating. Similarly, row 4, which has a missing value in the Favorite Color column, would be excluded from analyses involving Favorite Color. However, both rows would still be included in analyses involving other variables, like gender.

The **MCAR** assumption is a necessary condition for unbiased estimates of large samples; however, biases may occur if the missing data mechanism is **MAR**. Moreover, the strength of its power to estimate coefficients is dependent on the correlation pattern existing among variables. However, care must be taken in small samples, where the covariance matrix can be indefinite and therefore partly determinate, which gives rise to estimation issues [19].

### 2.4.3 Entire variables-deletion

Another possible approach is to remove the entire variable (*column*) from the analysis. Although there are no strict guidelines dictating when this step should be taken, it may be considered in scenarios where a substantial portion of data is missing, such as 60% or

more, and the variable is considered insignificant. In such cases, excluding the variable could be a viable option if it was inappropriately ignored (*see figure 2.3*) [19].

In this dataset, if we apply an entire variable deletion for Favorite Color due to the

Table 2.3: Example about entire variables-deletion [19]

ID	Gender	Depression Rating	Favorite Color
1	Male	6	Blue
2	Male	2	Green
3	Female	1	Red
4	Male	4	NULL
5	Female	5	Yellow
6	Female	9	NULL
7	Male	3	NULL
8	Female	4	Blue
9	Female	5	Blue
10	Male	8	NULL

presence of NULL values, we would exclude the Favorite Color column from any analysis or calculations involving that variable. This means that we would disregard any insights or patterns related to Favorite Color in the dataset.

Although excluding a whole feature (*column*) from the analysis certainly would make the process of analysis smoother, simpler, reduce noise, and address the issue of missing data, it also has the potential of losing vital information and introducing bias. Therefore, the exclusion of the variable should not be based only on its missing data level. Finally, the decision to leave out a variable is better made judiciously since it requires weighing the risk of oversimplification with the loss of information to reach an accurate and reliable outcome.

## 2.5 Missing data imputation

Instead of ignoring missing data, this class of methods focuses on a rough estimate of missing values. Different techniques have been deployed to calculate the missing values of the datasets. In what follows, we will present these imputation techniques.

### 2.5.1 Statistical techniques

Statistical metrics, such as mean, median, and mode, are extremely valuable tools employed in the field of missing data as well as in the fields of data analysis and machine learning to handle the missing values in datasets. Missing value imputation is one of the ways to manage these missing values by filling them with the mean (*average*), median (*middle value*), or mode (*most frequent value*) of the other data in the column [47]. These

techniques are presented below.

**Mean imputation:** A single imputation involves filling in the blanks with apparent data, and the reconstructed complete dataset is used in inference. Mean imputation (*MI*) is one such method in which it computes the mean for each variable and uses this mean value for imputing the missing values for that variable (see figure 2.2) [48].

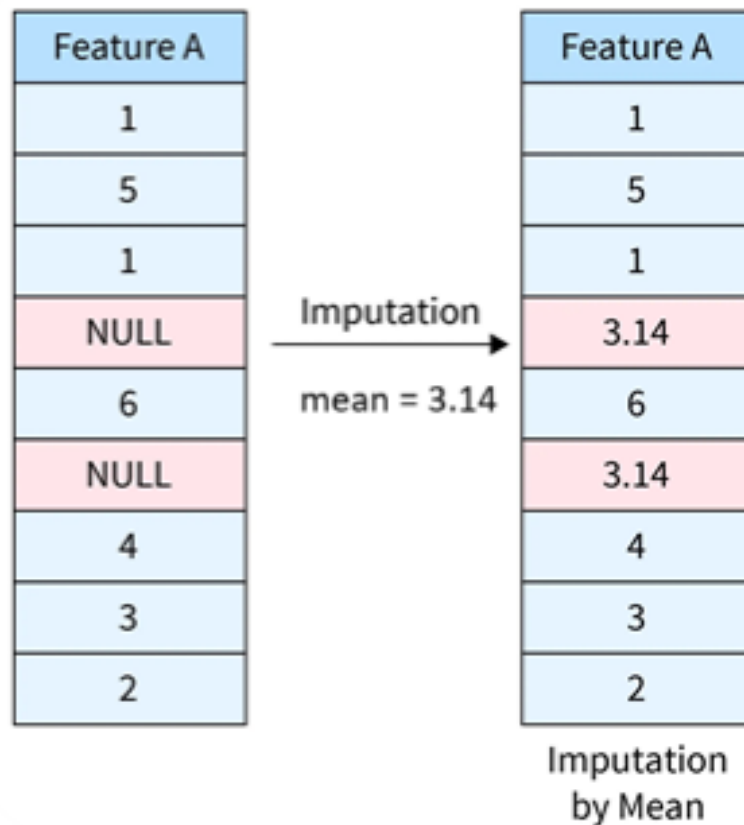


Figure 2.2: Example of mean imputation (MI) [10]

In the example of table 2.2, we handle missing data in the Feature A column, rows 4,6 using the mean metric.:

1. we calculate the mean of the non-missing values in the Feature A column: Mean =  $(1 + 5 + 1 + 6 + 4 + 3 + 2) / 7 = 3.14$ .
2. we replace the missing value (*NULL*) with the calculated mean: 3.14.

This method can give us other types of severely biased estimates, even if the data are **MCAR**. In fact, if the number of missing values in a variable is high and these values are based on the sample mean, then it is quite likely that the variance estimate for that variable will be underestimated.

**Median imputation:** The approach is to eliminate the outlier problems in the imputation



by the mean by replacing the NULL values with the column's median. This method holds only for numeric characteristics and ignores associative correlations (*see tables 2.3*) [19].

S.NO	Values
1	5
2	NULL
3	17
4	10
5	8
6	6

After Imputation →

S.NO	Values
1	5
2	8
3	17
4	10
5	8
6	6

Median = 8

Figure 2.3: Example of median imputation [11]

In the example of table 2.3, we handle missing data in the Values column, row 2 using the median metric.:

1. First we calculate the median of the non-missing values in the Values column:  
Non-missing values: [17, 10, 8, 6, 5] median = 8
2. Second we replace the missing value (*NULL*) with the calculated median:8

While median imputation is straightforward to implement and imposes a minimal computational burden, it carries the risk of information loss and potential bias. This risk escalates when missing values deviate notably from symmetry or fail to occur randomly within the dataset.

**Mode imputation:** Such approach is concerned with replacing missing or NULL values with the most recurrent value in the column. It comes into play just in the case of both numerical and categorical features. Yet again, as with other previous strategies, it disregards the feature correlation issues (*see tables 2.4*) [48].

**Example 6.** In the example of table 2.4, we handle missing data in the make column, row 7,10 using the mode metric.:

1. We calculate the mode, which is the most frequent value in the make column: Mode = Ford.
2. Replace the missing values (*NULL*) with the calculated mode: Ford.

As it's observed in the previous example, the mode imputation method is a fast and

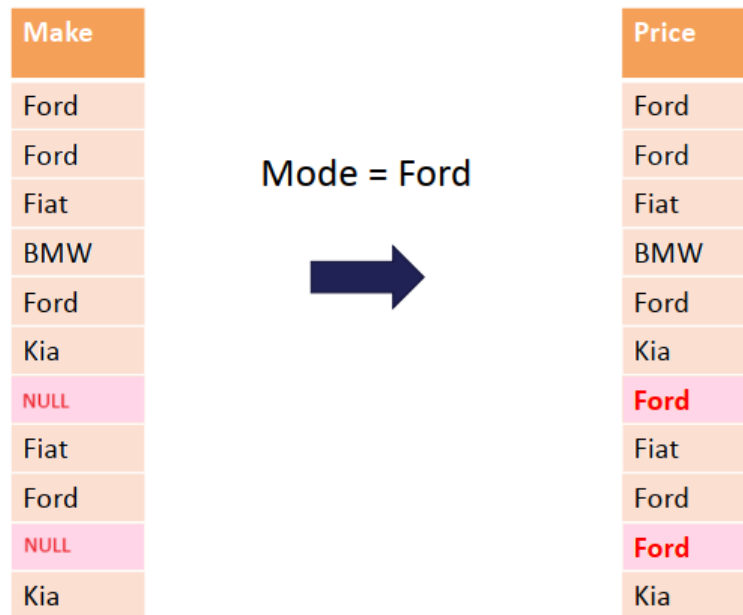


Figure 2.4: Example of mode imputation [12]

effective way to manage missing data in categorical variables, especially if the missingness is considered completely random. However, this technique fails to take into account the correlations between attributes within the dataset and looks for potential biases in skewed classes.

To overcome these limitations, we introduce below predictive models for data imputation that are more efficient than statical ones.

### 2.5.2 Predictive models

Missing data is replaced by some predicted values within the imputation process. Typically, the available non-missing data is utilized to forecast or estimate the values that will replace the missing ones. Below, we discuss some commonly employed imputation techniques based on predictive models.

**A) Regression imputation:** Among the common methods for imputing missing data, we find the linear regression model, which is built by using complete cases as training data, choosing appropriate predictor variables, fitting the model, and then using it to predict missing values based on the observed data.

#### a) Simple linear regression

Simple linear regression is a powerful tool for imputing missing data by exploiting the relationship between variables. It involves modeling the dependent variable with missing values using a related independent variable with complete data. The regression equation

is utilized:

$$Y = \beta_0 + \beta_1 X + \epsilon \quad (2.1)$$

Where  $Y$  represents the variable with missing values,  $X$  is the related variable,  $\beta_0$  is the intercept,  $\beta_1$  is the slope, and  $\epsilon$  is the error term.

Equation 2.1 represents the regression of the dependent variable  $y$  on the independent variable  $X$ . It enables the prediction of an unknown  $y$  value based on a given  $X$  value, by the equation:

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 X \quad (2.2)$$

Where  $\hat{y}$  represents the predicted value of  $y$ ,  $\hat{\beta}_0$  is the intercept estimate,  $\hat{\beta}_1$  is the slope estimate obtained from the simple linear regression model, and  $X$  is the independent variable value for which the prediction is made (see figure 2.5).

By leveraging this equation, we can estimate the value of  $y$  based on the given independent variable  $X$ , providing a means to predict missing or unknown values of the dependent variable in the context of simple linear regression [49, 50].

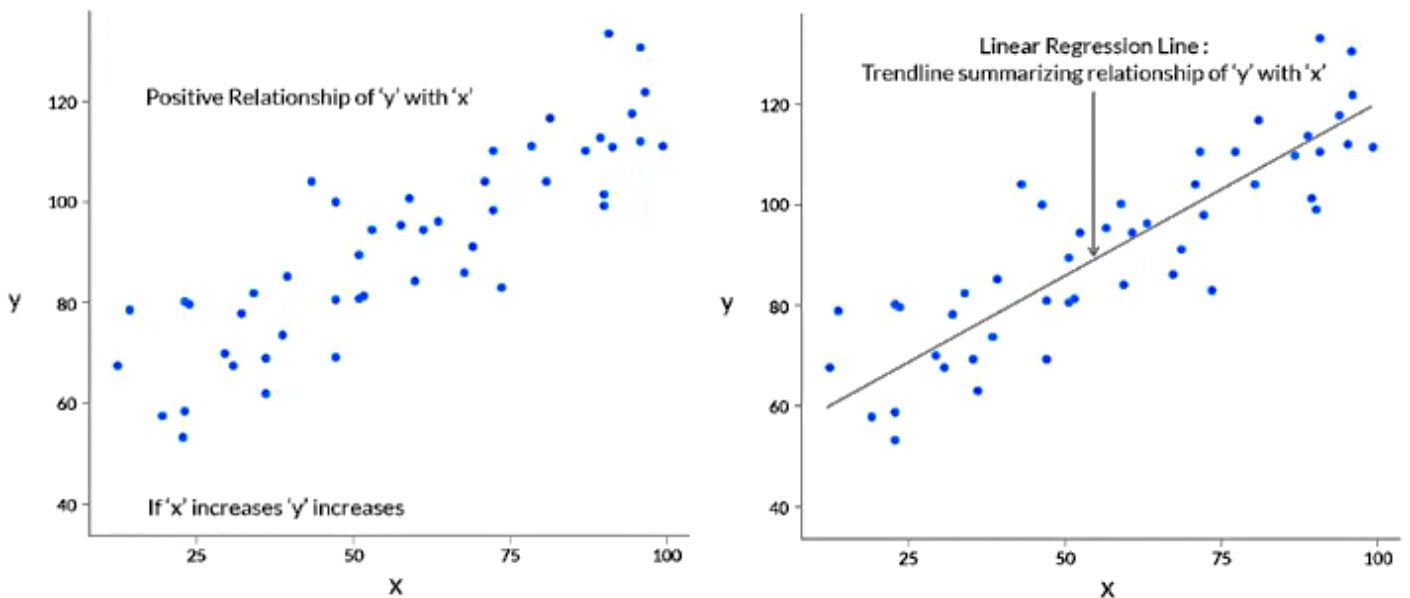


Figure 2.5: Simple linear regression plot [13]

**Example 7.** *The following example illustrates the regression technique for data imputation:*

*We'll use simple linear regression with Years of Experience as the independent variable and Salary as the dependent variable to impute the missing salary values.*

Table 2.4: Dataset: years of experience, education level, age, and salary before imputation

Years of Experience	Education Level	Age	Salary
1	Bachelor's	25	30000
2	Master's	28	35000
3	PhD	32	Missing
4	Bachelor's	30	40000
5	Master's	NULL	45000
6	PhD	35	55000
NULL	Master's	38	60000
8	Bachelor's	40	65000
9	PhD	45	70000
10	Bachelor's	50	75000

The process involves:

1. Fit a linear regression model using the observed Years of Experience and Salary values.
2. Use the fitted model to predict the missing Salary values for observations where Years of Experience is available.
3. Replace the missing Salary values with the predicted values, As in the table 2.5.

Table 2.5: Dataset: years of experience, education level, age, and salary After imputation

Years of Experience	Education Level	Age	Salary
1	Bachelor's	25	30000
2	Master's	28	35000
3	PhD	32	45000
4	Bachelor's	30	40000
5	Master's	27	45000
6	PhD	35	55000
7	Master's	38	60000
8	Bachelor's	40	65000
9	PhD	45	70000
10	Bachelor's	50	75000

This approach assumes a linear relationship between Years of Experience and Salary, allowing us to estimate the missing salary values based on the observed data.

### b) Multiple nonlinear regression

Multiple nonlinear regression is a method for filling in missing data that takes into account several interconnected variables and complex connections. It extends the simple linear regression idea by accounting for non-linear relationships and involving numerous

independent variables. This technique generates a model utilizing nonlinear functions to show the link between the dependent variable with missing values and several connected independent variables. The regression equation takes the form:

$$Y = f(X_1, X_2, \dots, X_k; \beta_1, \beta_2, \dots, \beta_p) + \epsilon \quad (2.3)$$

Where  $Y$  represents the variable with missing values, of the independent variables  $X_1, X_2, \dots, X_k$ , and  $\beta_1, \beta_2, \dots, \beta_p$  are the regression coefficients,  $f(\cdot)$  is a nonlinear function of the independent variables and regression coefficients, and  $\epsilon$  represents the error term. By fitting a nonlinear regression model to the data with complete information, the parameters  $\beta_1, \beta_2, \dots, \beta_p$  are estimated. These parameters capture the nonlinear relationship between the variables and are then used to predict missing values of the dependent variable based on the observed values of the related independent variables.

Imputing missing data using this method considers the intricate connections and nonlinear relationships between variables. This allows for accurate estimation of missing values, improving the completeness of the dataset without compromising the inherent relationships between its variables [49, 50].

**Example 8.** *We can use the previous example of table 2.4 in simple linear regression.*

1. *Years of Experience, Education Level, and Age are independent variables (features).*
2. *Salary is the dependent variable, with some missing values.*

*We'll use multiple nonlinear regression to impute the missing salary values based on the relationship between the independent variables (Years of Experience, Education Level, and Age) and the dependent variable Salary.*

This method reliably handles missing data, but it depends on assumptions like linearity and normality of residuals. If missing data patterns are more intricate or these assumptions don't hold, more advanced techniques may be necessary for precise analysis.

## **B) Imputation methods based on artificial intelligence**

Imputation techniques based on machine learning are advanced methods that typically leverage predictive modeling to address missing values using either unsupervised or supervised learning approaches. Like other imputation methods, these techniques rely on the available information from non-missing values in the data, utilizing labeled or unlabeled data to estimate the missing values.

Generally, when the available data contains valuable information for handling missing values, machine learning-based imputation methods can achieve high predictive accuracy (see figure 2.6).

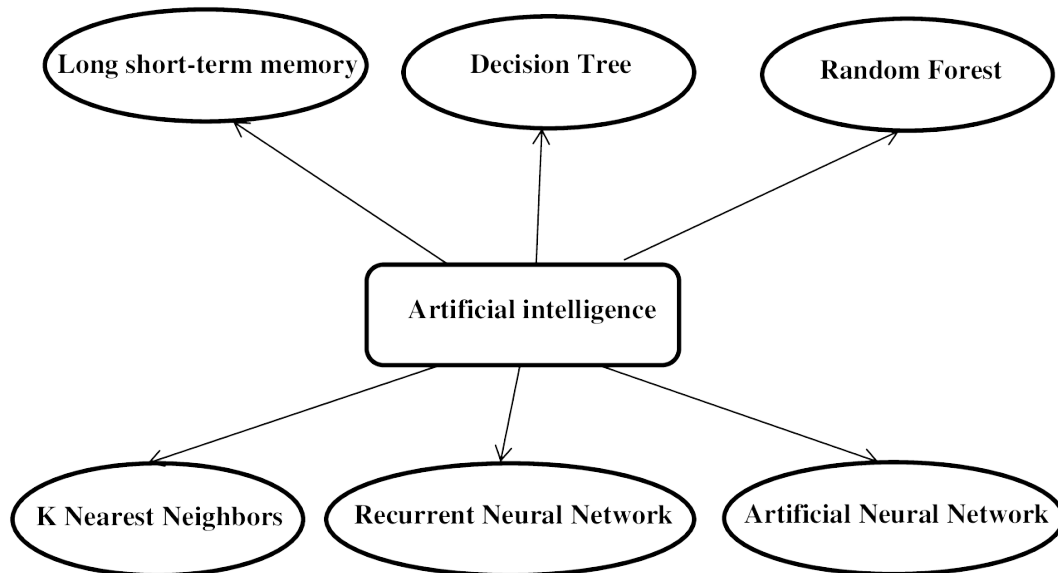


Figure 2.6: Artificial intelligence methods for missing data imputation

Below, we delve into some of the most extensively exploited machine learning imputation techniques [51].

### Using neural networks for handling missing data

In recent years, there has been a notable emergence of new predictive algorithms for survival, leveraging the power of neural networks (NN), particularly for individual prediction tasks, such as Recurrent Neural Networks (RNNs) and artificial neural network (ANN) [52], provide alternative approaches to handling missing data. In what follows these approaches are deeply discussed:

**a) Artificial neural network (ANN) :** Imputation algorithm based on a neural network (ANN) operates through several stages (See figure 2.7) [53].

- Initially, the dataset, comprising classification-type data with missing values, is prepared. Using the complete dataset as training data, the ANN model is trained to learn the relationships between input features and their associated classification labels.

- Subsequently, the trained ANN model estimates probabilities for each record's classification into different categories. When handling records with missing values, appropriate replacements are selected from complete records based on these probabilities.
- Following imputation, the algorithm selects the record with the highest probability of correct classification as the imputed result.
- The efficacy of this method is evaluated by comparing it with conventional imputation techniques using metrics like classification accuracy.

We can take many examples of this model from the incomplete database of hospitals, universities, schools, etc.

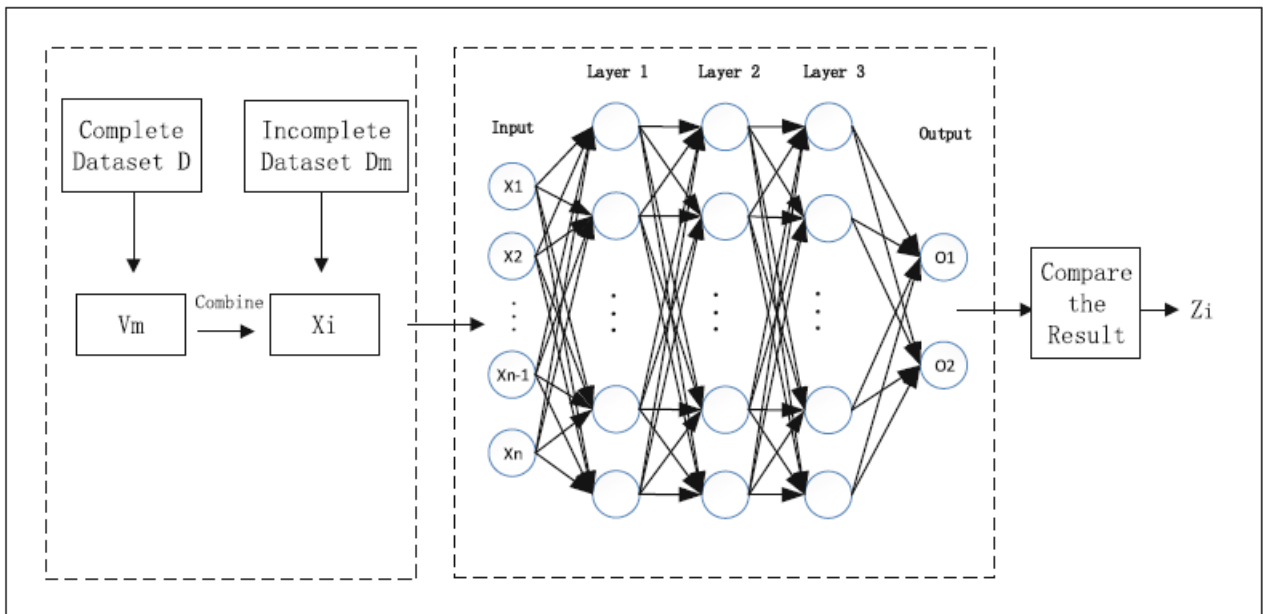


Figure 2.7: ANN model mechanism [14]

This approach harnesses the power of neural networks to effectively address missing data in classification scenarios, yielding superior classification results compared to traditional methods.

**b) Recurrent neural network (RNN):** Architecture integrates feedback connections among its units, particularly directing feedback to input units to estimate missing data. Initially, missing data are filled using mean imputation and then refined through feedback connections during the network's training for the classification task.

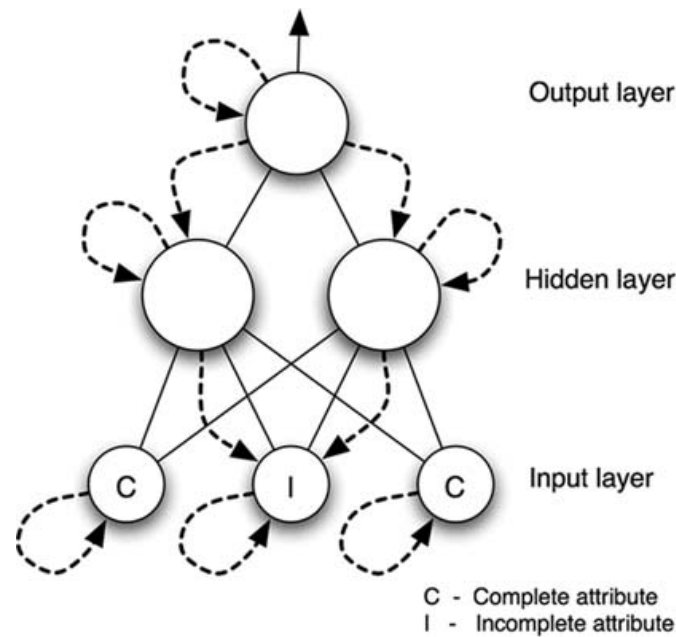


Figure 2.8: Missing data imputation with RNN [15]

This refinement process entails adjusting missing values according to the input missing in the previous iteration and the weighted combination of recurrent connections from other units (*both hidden and absent*) to the absent unit with a one-step delay. The diagram below visually represents this RNN methodology (*see figure 2.8*) [54].

Day	Sales (Units)		Day	Sales (Units)
1	150	RNN →	1	150
2	160		2	160
3	165		3	165
4	NULL		4	170
5	175		5	175
6	NULL		6	180
7	175		7	175

Figure 2.9: Imputing missing sales data with RNN

**Example 9.** In the example above (*see figure 2.9*) we have a time series dataset containing daily sales data value readings. However, there are missing sales values on some days, so we will use the RNN network to calculate these missing values.

Considering dependencies among input variables can improve output prediction; the recurrent network demonstrates superior performance compared to a standard network that replaces missing values with their mean.



c) **Long short-term memory (LSTM):** LSTM networks, a type of recurrent neural network (*RNN*), are designed to learn and remember long-term dependencies in sequential data. This capability makes LSTM particularly suitable for time-series data and sequences where the order and context of data points are crucial [55] (see table 2.6).

Table 2.6: Daily temperature readings with missing values

Day	Temperature (°C)
1	22.5
2	23.0
3	NaN
4	24.5
5	NaN
6	25.0
7	24.8

**Example 10.** In the table above (see table 2.6) we have an example of a time series dataset containing daily temperature readings over the course of a year. However, there are missing temperature values on some days so we will use the LSTM network to calculate these missing values.

Day	Temperature (°C)		Day	Temperature (°C)
1	22.5	LSTM →	1	22.5
2	23.0		2	23.0
3	NULL		3	22.0
4	24.5		4	24.5
5	NULL		5	23.5
6	25.0		6	25.0
7	24.8		7	24.8

Figure 2.10: Imputing missing temperature data with LSTM

This simple example demonstrates how LSTM networks can be used to impute missing values in a time-series dataset by leveraging the temporal dependencies and patterns within the data (see figure 2.10).

#### d) K nearest neighbour

The KNN algorithm operates by identifying the nearest neighbors of missing values and employing them for imputation through a distance metric that is calculated basing on the observed instances.

Various distance metrics, including Minkowski distance, Manhattan distance, Cosine distance, Jaccard distance, Hamming distance, and Euclidean distance, can be applied for KNN imputation. However, the Euclidean distance is commonly favored for its efficiency

and productivity, rendering it the most prevalent choice among practitioners. Below, we elaborate on KNN imputation employing the Euclidean distance measure.

$$\text{Dist}_{xy} = \sqrt{\sum_{k=1}^m (X_{ik} - X_{jk})^2} \quad (2.4)$$

Where  $\text{Dist}_{xy}$  is the euclidian distance,  $k$  represents the number of data attributes.  $j = 1, 2, 3 \dots k$   $k$  data dimensions,  $(X_{jk})$  : value for  $j$  attribute containing missing data, and  $(X_{ik})$  is the value of attribute containing complete data.

KNN imputation is a versatile technique that can handle various data types and multiple missing values.

Nonetheless, it has some limitations. It may not be as accurate in imputing values, and it might create false relationships between variables. Moreover, it requires searching through the entire dataset, resulting in longer computation times, making it less efficient [56].

**Example 11.** *For example, let's take a dataset that contains three attributes: age, income, and education level. Some values in the age column are missing, and we want to impute them using the k-NN algorithm.*

Table 2.7: Predicting missing data using KNN

Person	Age	Income	Education Level
A	25	50000	High School
B	30	70000	Master's
C	NULL	60000	Master's
D	40	80000	Master's

*To handle missing data using k-NN, we begin by preprocessing the dataset to manage missing values and outliers effectively. We then select relevant features like age, income, and education level, followed by normalizing the data to ensure consistent feature contributions.*

*Applying the k-NN algorithm involves identifying the k nearest neighbors based on available feature values, calculating distances, and imputing missing values by averaging values from these neighbors, thereby smoothly completing the imputation process.*

**e) Decision tree**

The decision tree algorithm in machine learning represents all possible outcomes and the associated paths leading to those outcomes in a tree-like structure. Utilizing this method for missing value imputation involves constructing decision trees to examine the missing values of each variable. Subsequently, the missing values of each variable are filled by utilizing its corresponding decision tree. The prediction for missing values is revealed in the leaf node of the tree. As the example below shows about handling budget data, (*see figure 2.11*):

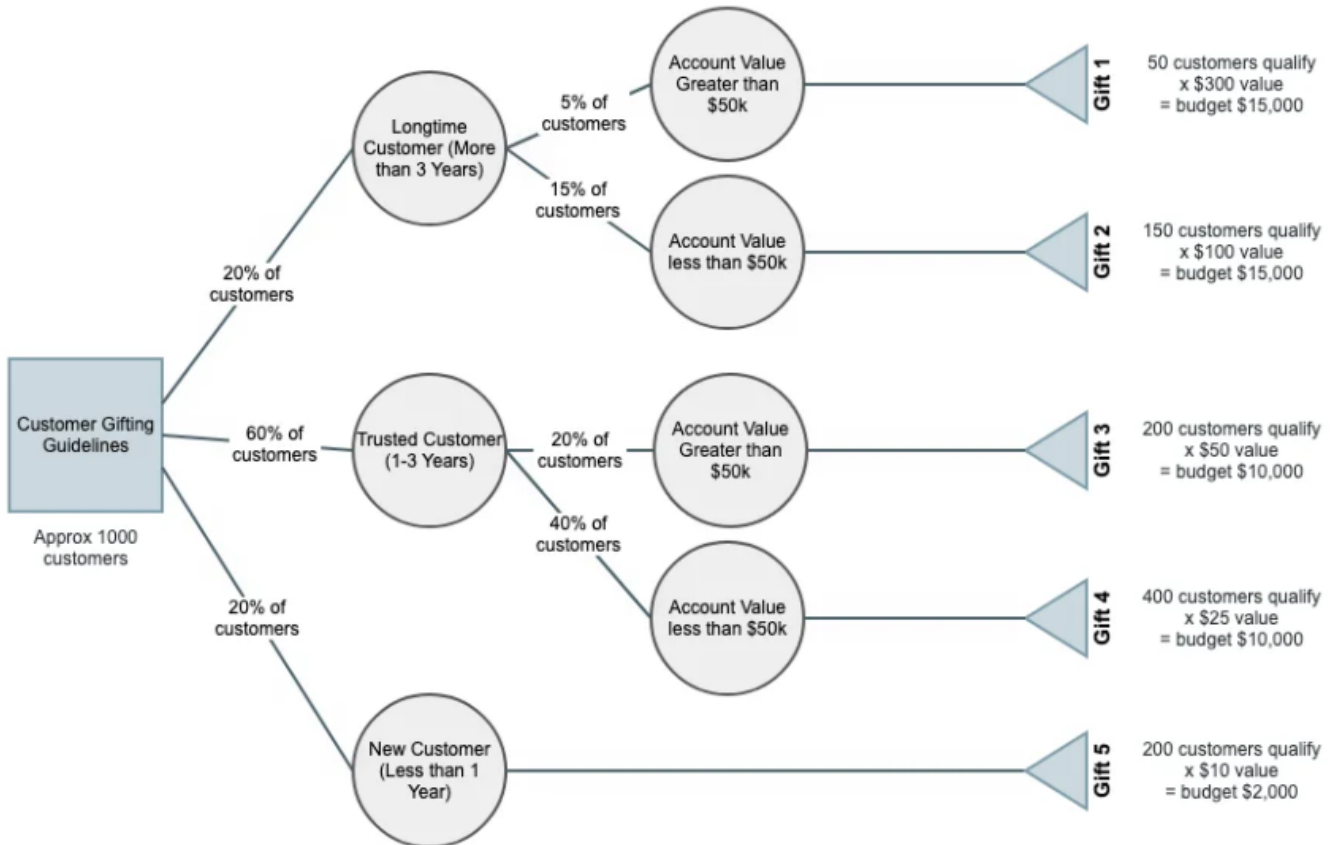


Figure 2.11: Predicting budget data using a decision tree [16]

This method can manage both numerical and categorical variables. It chooses important features and ignores unimportant ones. Even though decision trees are often complex, they are accurate because they have low bias in their predictions.

**f) Random forest**

A random forest consists of numerous trees, each functioning as a decision tree within a set learning framework. As an input sample enters the random forest, every decision tree evaluates and classifies it. Subsequently, through a process of scoring and assessment, the sample is categorized based on the majority decision among the trees in the forest [57].

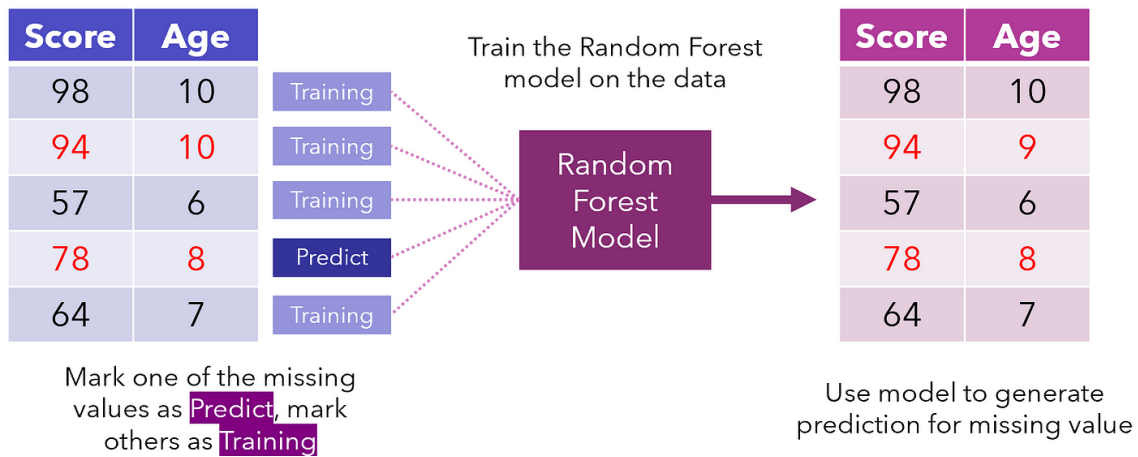


Figure 2.12: Example of random forest [17]

In the example shown in figure 2.12, the dataset is divided into two parts: training data and containing observed variables and missing data, used for prediction, so these sets are inputted into a Random Forest algorithm for the imputation of missing values. This process iterates until a stopping condition is met, ensuring continual improvement in data quality with each iteration. Typically, 5–6 iterations are sufficient to attribute the data accurately [58].

Random Forest is effective for handling missing data due to its ability to handle high dimensional datasets, maintain predictive accuracy, and utilize set learning to impute missing values robustly while minimizing bias.

## 2.6 Conclusion

Handling missing data is a critical challenging issue in data science area and the research literature is very rich in techniques that deal with this issue. In addition to basic statistical techniques, the imputation of missing data is a complex task that requires careful consideration. There are a plethora of methods available, each of which is characterized by its own strengths and limitations. There are no one-size-fits-all solutions, and statisticians often need to tailor their approach to the specific characteristics of each dataset.

Furthermore, the process of handling missing data often involves iterative steps, requiring statisticians to navigate between the raw data and the corrected or imputed data.

This method can manage both numerical and categorical variables. It chooses important features and ignores unimportant ones. Even though decision trees are often complex, they are accurate because they have low bias in their predictions.

The next chapter is dedicated to related works that have tackled the issue of missing data.

Chapter **3**

Related work

### 3.1 Introduction

The issue of missing data is a significant challenge in various fields, including healthcare, finance, social sciences, and more. The prevalence of missing data across these diverse domains underscores the critical need for effective strategies to handle and mitigate its impact on data analysis, decision-making, and research outcomes. Many isolated projects are being conducted worldwide on the theme of missing data, highlighting the importance and complexity of addressing this issue. The objective of this chapter is to provide an overview of the diversity of concepts and goals addressed by different works related to missing data. In order to examine the work dealing with missing data, we start by exposing our research methodology.

### 3.2 Presentation of the research methodology

The figure 3.1 below shows the adopted research methodology.

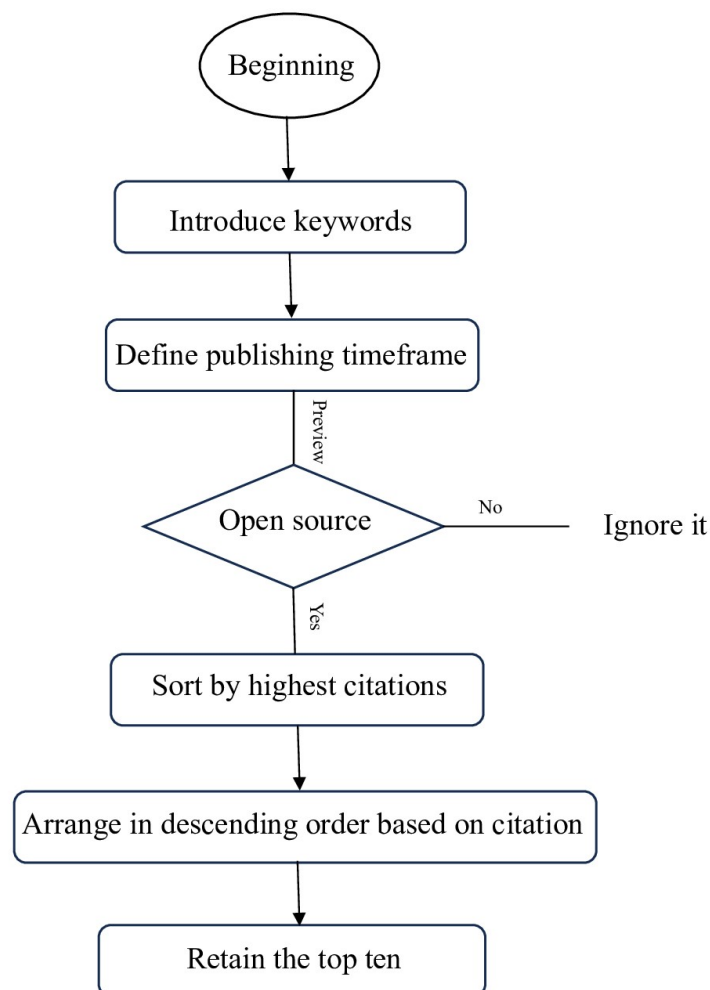


Figure 3.1: Methodology model

For exploring the research literature by seeking related works for our report on missing data, we employed a keyword approach using significant terms, such as: **missing data, missing value, incomplete data, imputation, artificial intelligence, machine learning, deep learning, MCAR, MAR, MNAR, deletion, handling, missing completely at random, missing at random, missing not at random, mechanisms, and selection bias.** Alongside the keyword approach, we also used citation tracking and reference chaining to find more relevant literature on missing data. By combining these methods and filtering out older articles and non-open-source materials, we were able to conduct a focused and comprehensive exploration of the available literature.

This approach facilitated a thorough investigation of the literature on missing data and included a variety of relevant articles for detailed analysis and review.

### 3.3 Selection process and article acquisition overview

The research process is based on using Google Scholar. We formulated incrementally refined requests and submitted them to the search engine. The initial request was based on the unique keyword 'incomplete data'. Unfortunately, this yielded an overwhelming number of results, with over **9,140,000** resources.

Subsequently, we refined our search by adding the rest of the keywords, this refinement resulted in a decreased number of results with **542** resources retrieved. However, this number was still considered relatively large.

In order to narrow down the results to more relevant ones, we refined the search to include only documents developed since 2021. Hence, this refinement resulted in a significant difference in the number of results. Initially, with the first request using the keyword 'incomplete data', we obtained **29,400** results.

Subsequently, after adding all keywords, the number decreased to **392** articles, indicating a noticeable drop in search results compared to the previous results. This approach ensured that older studies were excluded from our analysis (*see table 3.1*).

Further refinement by considering only open-source articles resulted in **112** relevant articles, ensuring access to freely available and transparent research materials. From this pool, we arranged the articles in descending order by number of citations, and then we selected the first **10** articles, which not only represent a comprehensive overview but also provide insights into the most impactful contributions in the field.

These articles form the basis of our review and analysis, allowing us to delve deeply into various approaches and techniques for dealing with missing data.

Table 3.1: Google scholar results for missing data related keyword phrases since 2021

#	Key words	Number of results
1	incomplete data	29,400
2	1 + missing data	17,900
3	2 + MCAR	17,300
4	3 + deep learning	16,900
5	4 + machine learning	16,700
6	5 + artificial intelligence	16,400
7	6 + MAR	11,700
8	7 + MNAR	870
9	8 + deletion	558
10	9 + imputation	534
11	10 + handling	533
12	11 + missing at random	519
13	12 + missing completely at random	519
14	13 + missing not at random	519
15	14 + missing value	519
16	15 + mechanisms	443
17	16 + selection bias	392

Through meticulous curation of pertinent literature, It becomes necessary to analyze the collected 10 works.

## 3.4 State of the art analysis

Analyzing current research and methodologies reveals complexities and offers insights into the strengths and limitations of existing approaches for paving the way for informed decision-making in our study or project.

### 3.4.1 Related works analysis

These projects encompass a wide range of topics, showcasing the breadth of our interests and expertise. They span across fields such as technology, education, healthcare, and environmental sustainability, reflecting our commitment to addressing multifaceted challenges related to missing data.

-In [59], Jäger et al. emphasize the detrimental impact of missing values on data pipelines and downstream ML applications. This research carries out experiments on diverse datasets with varying missing scenarios. The datasets used in the experiments were obtained from the OpenML database. Key findings reveal that simpler supervised learning methods like k-NN, random forest, and discriminative deep learning approaches often outperform modern generative deep-learning-based methods in MCAR and MAR settings. Additionally, mean/mode imputation performs well for categorical columns in



challenging imputation scenarios.

-the authors in [60] introduce a solution for missing data in structural health monitoring systems with wireless sensors. They propose a novel data-driven GAN approach that overcomes the limitations of traditional correlation-based imputation methods, especially for complex correlations like vehicle-induced strains. Despite a low sampling frequency of 1 Hz, their method effectively distinguishes dynamic from static responses, showcasing practical utility and efficiency. They also discuss scalability to higher frequencies, noting longer training times but highlighting adaptability to different monitoring scenarios.

-In [61], Johnson et al. address the challenge of missing data in ecological and evolutionary phylogenetic comparative studies. The study simulated continuous traits and response variables to test nine imputation methods and complete-case analysis under biased missing data scenarios. Results indicate that while Rphylopars imputation generally provides accurate estimates and preserves response-trait relationships well, none of the tested methods effectively handle the severe biases common in trait datasets.

-In [62], Bähr et al. study delves into the quality of geolocation sensor data from smartphones and identify various error sources affecting geolocation data accuracy. The study uses a multi-step error model to analyze and reduce these errors, providing insights into the complexity of working with sensor data in social science research. Their study revealed challenges like gaps in data collection due to device differences, user actions affecting data quality, and the influence of smartphone hardware and operating systems on location measurements.

-In [63], Izonin et al. propose an improved set method using two GRNNs and an extended-input SGTm neural-like structure to manage missing data in smart systems. The method shows superior accuracy in recovering missing or lost data in real air conditioning monitoring datasets compared to existing methods. The study confirms the benefits of blending methods to forecast data using artificial neural networks. It emphasizes the important role this approach will play in future research on data management and prediction. The findings demonstrate the effectiveness of this combined technique for making accurate data predictions.

-The authors in [64] present a cloud-based system with advanced algorithms for real-time imputation of missing data in high-frequency water quality monitoring. They evaluate ten imputation methods using water temperature and nitrate concentration data from monitoring systems. Their findings highlight the Dual-SSIM method as excelling in accurately imputing missing values by effectively leveraging temporal patterns. They also highlight the limited effectiveness of basic approaches like replacing missing values with the average, especially when dealing with intricate data variations. The study also examines the effect of data gap size on the accuracy of imputation; it highlights the strengths of neural network-based methods, especially when dealing with larger data gaps.

-The authors in [65] investigate the prevalence of missing data and its impact on overall survival among cancer patients using the National Cancer Database for non-small cell lung cancer, breast cancer, and prostate cancer. Substantial proportions of patients in these cohorts had missing data, ranging from 39.7% to 71.0%. Patients with missing data generally experienced lower 2-year overall survival rates compared to those with complete data. These results underscore the vital importance of enhancing data documentation and quality within clinical registries, and this is essential to gain deeper clinical insights and ultimately enhance patient care outcomes.

-The authors in [66] propose CI-clustering and LI-clustering algorithms for density-based clustering with incomplete data, overcoming the limitations of traditional imputation methods. CI-clustering conducts imputation and clustering concurrently based on Bayesian theory, while LI-clustering addresses low-density areas in clusters through local imputation. The experimental results show that both algorithms are effective and suggest new directions for future research in managing large datasets and data streams.

-In [67], Mutasim et al. address the critical concern of data quality in machine learning and related fields, highlighting the challenge posed by missing values. To tackle this issue, the authors propose an imputation approach using the K-Nearest Neighbor (KNN) algorithm with IBK classification in R. The imputed data, saved as "imputed.csv" undergoes comparison with the original dataset to validate the imputation process. The study shows that the proposed method adeptly handles missing data, offering researchers complete datasets for in-depth analysis.

-In [68], Kim et al. address the issue of missing values in marine fleet data by proposing a regression-based method for estimating ship principal data. Through a case study involving 6,278 container ships, the method demonstrated a significant improvement of up to 15.6% in accuracy compared to previous techniques. The model's effectiveness was also observed in handling ships with dimensional constraints, meeting standards for canal passages.

Table 3.2: Summary of existing work focusing on missing data

Researchers	Year	Title	field of application	limitations / remarks	Evaluation Metrics/Performance Measures
[59]	2021	A benchmark for data imputation methods	Data engineering, database management systems (DBMSs), and machine learning (ML) applications	Imputation methods are limited by the specificity of the dataset and evaluation metrics	Root mean square error (RMSE) and macro F1-score
[60]	2022	Continuous missing data imputation with incomplete dataset by generative adversarial networks-based unsupervised learning for long-term bridge health monitoring	Structural health monitoring systems	Scalability issues with larger datasets and increased errors in multiple-sensor imputation under-score	Root mean square error (RMSE) and recovery error
[61]	2021	Handling missing values in trait data	Ecological and evolutionary phylogenetic studies	The inability of tested methods to handle severe biases effectively	Root mean square error (RMSE)
[62]	2022	Mobile geolocation sensor	Geolocation sensor data quality analysis in social science research and analytics	The study's scope did not include investigating the effects of GPS falsifier apps, device sharing, or machine learning for user identification	Not mentioned
[63]	2021	An approach towards missing data management using improved GRNN-SGTM set method	Smart systems particularly in air condition monitoring	Computational resource requirements and potential sensitivity to dataset variations warrant further investigation for broader applicability	MAPE (Mean Absolute Percentage Error) and RMSE (Root Mean Square Error) and operating time
[64]	2022	Handling missing data in near real-time environmental monitoring: A system and a review of selected methods	High-frequency water quality monitoring systems	computational complexity, potential biases, data dependency, interpretability challenges, and the need for hyperparameter tuning in described imputation techniques	Root mean square error (RMSE) and mean absolute error (MAE)
[65]	2021	Prevalence of missing data in the national cancer database and association with overall survival	Cancer registries	Incomplete outcomes assessment, data abstraction variability, heterogeneous study populations	Not mentioned
[66]	2021	Effective density-based clustering algorithms for incomplete data	Handling incomplete datasets in various real-world scenarios	Challenges with large incomplete datasets and diverse cluster shapes	Not mentioned
[67]	2021	Impute Missing Values in R Language using IBK Classification Algorithm	Artificial Intelligence (AI) and Machine Learning (ML), Data Management and Cleansing, and Missing Data Handling in Machine Learning	The study is limited to k-NN imputation and IBK classification, necessitating future exploration of diverse imputation methods, additional metrics evaluation, scalability concerns, and automated missingness mechanism investigation	Not mentioned
[68]	2022	A novel method for estimating missing values in ship principal data	The fleet dataset in the marine sector	Sample size dependency, limited generalizability, scope of model comparison, need for industry validation, and potential for model improvement	MAE (Mean Absolute Error) and RMSE (Root Mean Squared Error) and MSE (Mean Squared Error) and Adjusted R-squared

Despite numerous works addressing missing data using diverse techniques, persistent challenges remain evident.

### 3.4.2 Synthesis of related works

Although many works have been conducted to tackle the issue of missing data by adopting various techniques and approaches, it is observed that the following issues remain acute and require more rigorous approaches for their management.

#### a. Data variability and sensitivity

Data variability refers to the variety and complexity present in different datasets. Sensitivity relates to how well methods perform across varying data conditions, where articles [59], [63], [64] and [67] discuss the challenges of effectively handling dataset variations, ensuring methods are not overly sensitive to specific dataset characteristics, and the need for automated mechanisms to appropriately address missing data.

#### b. Imputation limitations

Imputation involves filling in missing values within a dataset, and this topic underscores the challenges and constraints linked with imputation techniques. Articles [59], [60], [61], [64], and [67] delve into issues like the limited capability of some methods to effectively manage severe biases, the dependency on particular data traits that may not exist universally, potential biases arising from imputation processes, and the necessity of employing a range of imputation methods combined with thorough metric evaluations to accurately gauge their performance.

#### c. Scalability issues

Articles [59], [63], [66], and [68] all mention issues such as increased errors when dealing with large amounts of data, the need for significant computational resources to process such data, and the complexity that arises when working with diverse cluster shapes within datasets, where scalability issues can manifest in various ways, including longer processing times, higher memory requirements, and difficulties in maintaining performance as the dataset size increases.

The review of related work highlights the fact that it becomes evident that various methodologies have been explored to address the identified research gaps. However, the proposed methods suffer from specific shortcomings. To better handle missing data, a new, more adequate approach for handling missing data must be considered. The new approach must benefit from recent advances in the field of artificial intelligence.

## 3.5 Conclusion

In summary, the review of relevant work on missing data underscores the complexity and continuing challenges of effectively addressing this problem, where different methodolo-

gies and techniques have been explored.

However, the lack of a universal solution highlights the need for context-specific approaches tailored to different datasets and missing data patterns. This sets the stage for the next chapter, where we will delve into the conception of a novel framework to handle missing data comprehensively.

## **Part II**

# **Modeling and implementation of the approach**

Chapter **4**

# The hybrid approach for handling missing data

## 4.1 Introduction

In this chapter, we will focus on developing a novel comprehensive framework for handling missing data. The target model consists of an hybrid approach that combines predictive models with deletion and imputation techniques. The ultimate goal is to allow users and managers to benefit from an advanced method that profits from both combined techniques. Thus, the proposed framework offers a broad spectrum of choices for dealing with incomplete information.

The content of the chapter deeply discusses the main features of the proposed approach. We start by presenting the framework features, then we expose the system architecture and functionalities. After that, we illustrate the system input and output, then we expose the enhanced techniques for managing missing data. The conceived framework is deployed in a real-world scenario in order to show its feasibility and applicability. Finally, we summarize the chapter with a conclusion.

## 4.2 Framework features

The framework to be conceived is characterised by the following features:

- **A) Combining a variety of approaches aimed at effectively addressing missing data:** The framework integrates a comprehensive set of techniques and AI algorithms designed to address different types and incomplete levels of missing data. This ensures strong data processing capabilities and more flexibility.
- **B) Offering a customizable approach that allows dealing with various scenarios:** Users can customize the framework be adapted to specific data environments and analytical requirements, which provides flexibility and versatility when effectively dealing with various missing data scenarios.
- **C) Providing the user with the possibility to choose the suitable technique for a specific usage context:** The framework enables users to choose the most suitable method based on the characteristics of managed data and the objectives of the analysis, which enhances accuracy in data processing.
- **D) Implementing the approach in a software tool that offers various functionalities in an ergonomic interface:** The framework is realized as a user-friendly software tool integrating a large range of functionalities designed to streamline the missing data processing workflow, offering an ergonomic interface for enhanced usability and efficiency.
- **E) Comparing the resulting values for various selected methods:** The framework enables users to identify the most effective and advantageous approach for their specific application by comparing the obtained outputs.



### 4.3 Areas of framework usage

In this section, we highlight the various domains that can benefit from the functionalities offered by our framework. As the framework is highly versatile, it can be customized to handle a wide variety of datasets originating from various application areas, such as resources management, energy security, food and healthcare fields. Particularly, the following domains can potentially benefit from the system.

- **Decision support systems:** Filling in the gaps with predicted values can improve decision support systems by providing more complete and accurate data for analysis and decision-making. Hence, it allows for a more informed decision.
- **Fraud detection:** Instead of using incomplete data, fraud detection algorithms rely on complete data to identify suspicious patterns.
- **Risk management:** In finance and insurance, this framework helps manage risk by ensuring that missing data does not lead to inaccurate risk assessments or underwriting decisions. Thus, it helps avoid losses of money and investments.
- **Customer relationship management:** The proposed framework can help Customer Relationship Management ([CRM](#)) systems by ensuring customer data is complete and up-to-date, leading to better customer segmentation, targeting, and personalized marketing strategies.
- **Critical systems:** Such systems are real-time and require accurate data to calculate values used for making decisions. Indeed, they manage human lives, material resources, and various critical infrastructures such as nuclear factories, dams, satellite systems, and space navigation.
- **Human resources management:** Human resources departments often deal with large sets of data related to employee performance. By calculating the missing values, the manager will be able to make the right decisions accordingly.
- **Manufacturing and quality control:** In manufacturing processes, missing data can affect quality control measures and production efficiency. By addressing missing data, our framework enhances the accuracy of quality control inspections, predictive maintenance models, and more.

## 4.4 Architecture of the proposed framework

The following model represents the general structure of the framework.

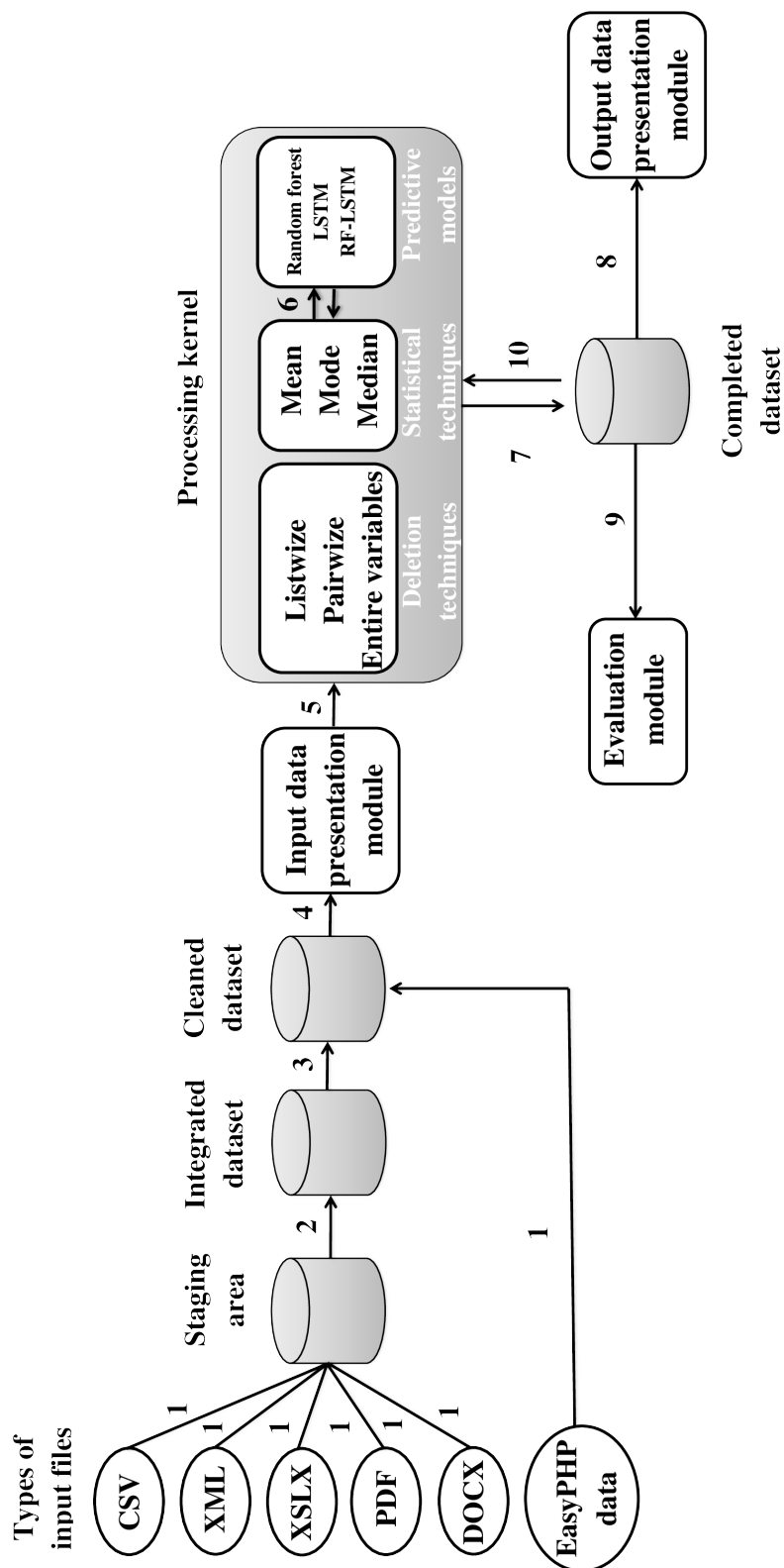


Figure 4.1: Architecture of the proposed approach

As observed in figure 4.1, the proposed architecture is articulated around a set of components that interact together to achieve the previously described features. These components and their interactions are further explained below.

#### 4.4.1 Components of the architecture

The architecture manipulates a variety of components, each serving a distinct purpose, so defining and clarifying them is crucial to understanding the overall framework.

Table 4.1: Description of the system components

Component	Description
Types of input files	One or more datasets of various types can be entered: <ul style="list-style-type: none"> <li>• Comma-Separated Values (CSV).</li> <li>• XML.</li> <li>• Microsoft Excel Open XML Spreadsheet (XLSX),</li> <li>• Portable Document Format (PDF).</li> <li>• DOCX.</li> <li>• EasyPHP dataset.</li> </ul> In order to process missing data staging area.
Staging area	Temporary storage containing the input datasets.
Integrated dataset	Contains all the input dataset after integration.
Cleaned dataset	A pre-processed merged dataset prepared to process its missing data.
Input presentation module	This module displays statistical values related to the input dataset.
Processing kernel	This is the most important component of the architecture as it represents the various methods that can be deployed by users to handle missing data in the combined dataset ( <i>deletion techniques, statistical techniques, and predictive models</i> ).
Completed dataset	This component shows the merged dataset that has undergone missing data processing, and that are free of any missing data.
Output presentation module	This module displays statistical values related to the output dataset, such as the percentage of missing data and more.
Evaluation module	This component depicts a comparison between previously chosen methods for handling missing data such as execution time and memory usage.

The conceived framework offers an efficient decision support tool. The interaction between the previous components are explained below.

### 4.4.2 Interaction between the system components

In this part, the various interactions allowing to progress from one stage to another during the handling missing data process are explained. This action is achieved by describing the numbered arrows in the figure 4.1.

1. The different types of datasets to be processed are uploaded from their respective sources and stored in the staging area.
2. After data storage, all datasets are integrated into a single dataset and presented to the user.
3. At this stage, the data is preprocessed. This activity includes removing duplicate records, encoding categorical features, and performing data cleaning.
4. The display module showcases statistics for the input datasets, such as the percentage of missing data, the number of features, and other metrics related to the input dataset..
5. At this stage, the data is ready to be processed. Hence, a particular method is selected from among the proposed ones to address missing data (see the next section 4.7).
6. To train our models, we rely on statistical methods to address any data deficiency for comprehensive training, and for integrating the two models, we also require them.
7. The resulting values obtained as outputs of the processing mechanism are stored in their respective positions (cells). This process allows for a more complex dataset.
8. The display module shows statistics for the output datasets, such as the percentage of missing data, the number of features, and graphs of the distribution of all data in a dataset before and after missing data processing.
9. For performance reasons and decision-making perspectives, the evaluation module presents a comparison of the results of the methods chosen for evaluation.
10. In cases where the obtained values are not satisfactory, the user can reiterate the process of missing data handling by choosing another method from the proposed ones.

## 4.5 Input / output of the conceived system

In this section, we present the various inputs and outputs taken into account by the framework. Understanding these elements is a crucial step towards deploying our framework while effectively managing the calculated missing values.

In fact, our framework is able to handle various input formats, ranging from [CSV](#), [XLSX](#), [XML](#), [PDF](#), [DOCX](#), or directly from EasyPHP (*see figure 4.2*).

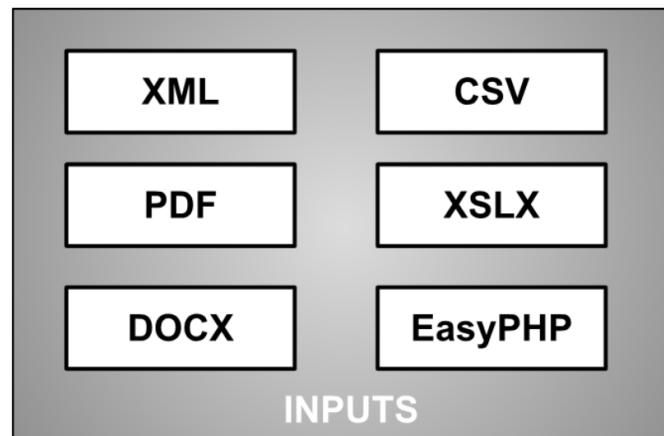


Figure 4.2: The various input formats

Upon processing the previous input data, the system generates the following resulting elements:

- Completed dataset that can be conveniently stored in [CSV](#) format.
- Statistical analyses derived from the integrated dataset.
- A comprehensive comparison of the methods utilized for handling missing data.

These elements are illustrated in the figure 4.3 below.

The motivation behind using [CSV](#) files as an output format is justified by the following

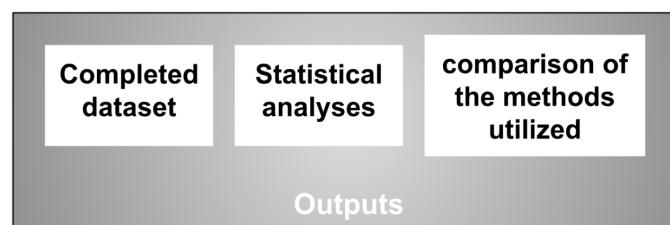


Figure 4.3: Framework outputs

reasons:

- [CSV](#) files are a commonly used format that is deployed by various communities and Electronic Data Interchange ([EDI](#)) areas.
- there exists a large panoply of soft tools for managing this format.
- [CSV](#) files are simple formats.
- [CSV](#) files are relatively compact compared to other data formats like XML or JSON, resulting in smaller file sizes and efficient use of storage space.
- [CSV](#) files are easy to inspect and modify if needed.

## 4.6 Dataset description

Throughout our illustration and during the system learning phase, we used the London weather dataset. This dataset was obtained from Kaggle includes 15,343 records and serves as the basis for training the three models (*Random Forest, LSTM, and hybrid model RF\_LSTM*) and testing them on the remaining methods (*statistical and deletion techniques*) described in 4.7.4, which contain various features cloud cover, sunshine, global radiation and more. Kaggle is a popular platform for hosting datasets and machine learning competitions, providing access to diverse datasets across various domains.

The weather dataset undergoes pre-processing to deal with missing values and ensure data integrity before training the model. The table below shows an excerpt of the dataset:

Table 4.2: An excerpt of london weather dataset

Date	Cloud Cover	Sunshine	Global Radiation	Max Temp	Mean Temp	Min Temp	Precipitation	Pressure	Snow Depth
19790101	2.0	7.0	52.0	2.3	-4.1	-7.5	0.4	101900.0	9.0
19790102	6.0	1.7	27.0	1.6	-2.6	-7.5	0.0	102530.0	8.0
19790103	5.0	0.0	13.0	1.3	-2.8	-7.2	0.0	102050.0	4.0

## 4.7 System functionalities

To understand the complex mechanism of the system in what follows we deeply explore it's functionalities.

### 4.7.1 Storing and merging datasets

This marks the primary stage wherein all input datasets are systematically collected and stored. It's imperative to note that the framework is designed to accommodate diverse types of datasets ([CSV](#), [XML](#), [XLSX](#), [PDF](#), [DOCX](#), EasyPHP dataset), which are subsequently merged to create one cohesive unified dataset in a CSV format (*see figure 4.4*).

In the example 4.4, after applying collection and integration (*sorting and merging*), we find:

- **Input:** The first dataset is in [XML](#) format, and the second is in [CSV](#) format.
- **Applying collection and integration (*sorting and merging*) gives:** A target dataset that contains the features of the first and second datasets is created, and if they have the same features, a number is added next to them according to the order of entry.
- **Output:** Merged dataset.

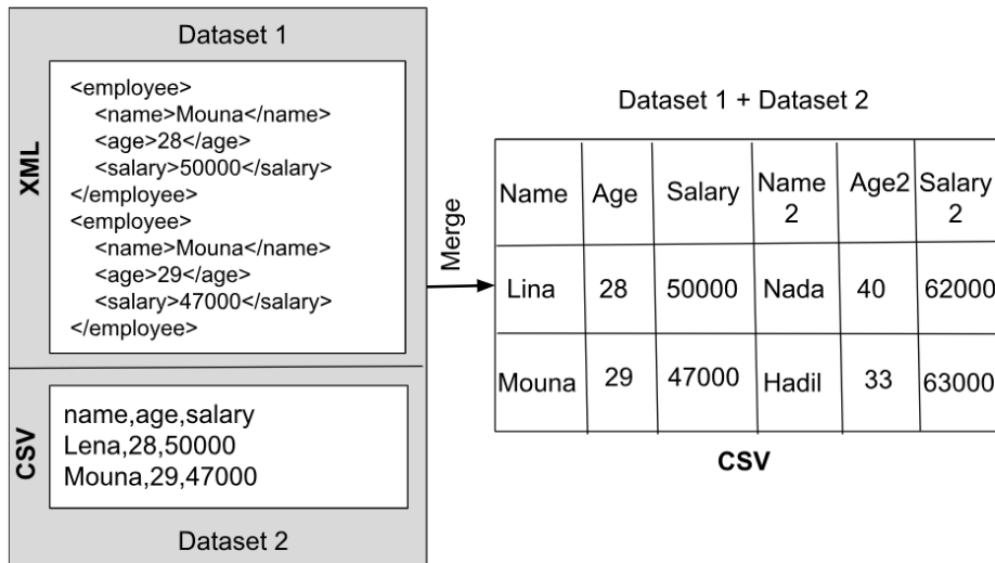


Figure 4.4: Example of storing

### 4.7.2 Detection and identification of missing data

At this stage, incomplete data stored in data sources is detected and identified based on a search of NaN (*missing data*) values in different attributes. This proactive approach not only detects but also efficiently highlights missing data from existing ones. Hence, such a process allows a more enhanced user clarity and comprehension.

The mechanism for detecting and identifying missing data is illustrated in the following example (*see figure 4.5*).

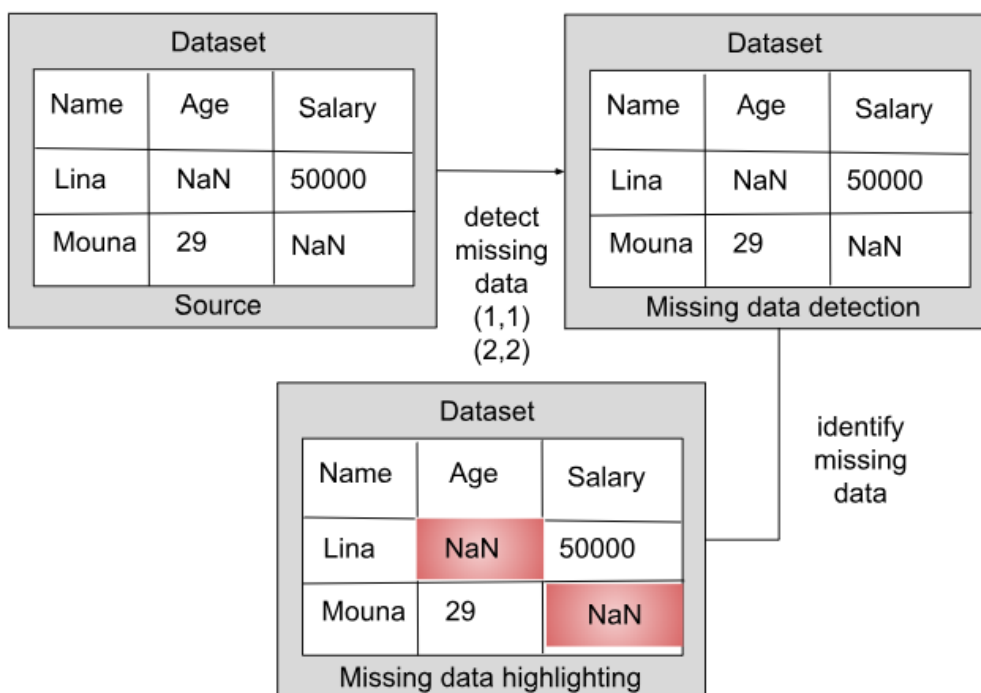


Figure 4.5: Example of detection and identification of missing data

In the example 4.5, we observe that applying detection and identification of missing data, we find:

- **Input:** Incomplete preprocessed merged dataset.
- **processing detection and identification of missing data:** The missing data was detected (*for example, in cells (1, 1), (2, 2)*), and then it was highlighted with a red color so that the user could distinguish it.
- **Output:** The dataset with its missing highlighted data.

### 4.7.3 Input dataset statistics

For an assessment of the quality of the data, this model allows for different statistics. Thus, the quality of the provided dataset is calculated and evaluated by providing statistical parameters related to the provided dataset, such as the number of features in the input dataset, percentages of missing data in different features, graphs of the distribution of missing data, and the distribution of all existing data as well (*see figure 4.6*).

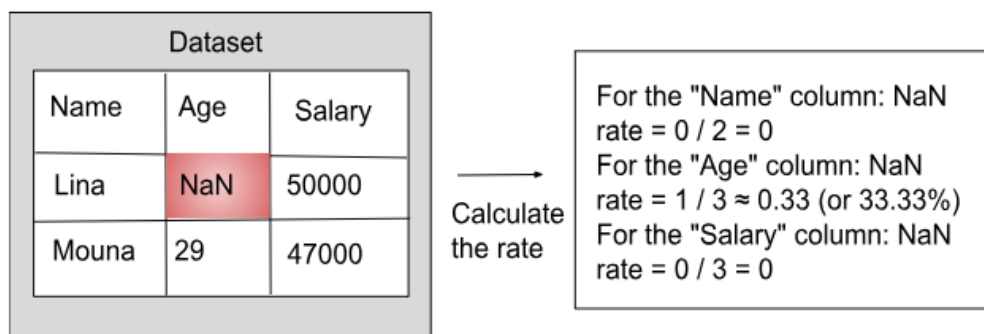


Figure 4.6: Example of input dataset statistics

In the example 4.6, after applying input dataset statistics, we find:

- **Input:** Incomplete pre-processed merged dataset that has missing data flagged.
- **Applying input dataset statistics gives:** The rate of NaN (*missing values*) in the dataset was calculated by counting the number of NaN values focusing on different criteria and then dividing by the total number of entries in that column. There are three types of average calculations: by **row** or **column** or **column and row**.
- **Output:** The rate of missing data in columns.

### 4.7.4 Missing data imputation

The proposed framework offers to system users various techniques for effectively managing missing data. Among these methods, we suggested deletion techniques, statistical techniques, and predictive models. When using one or more combined techniques, the



user will have the flexibility to select the most suitable approach for its specific context and requirements (see figure 4.7).

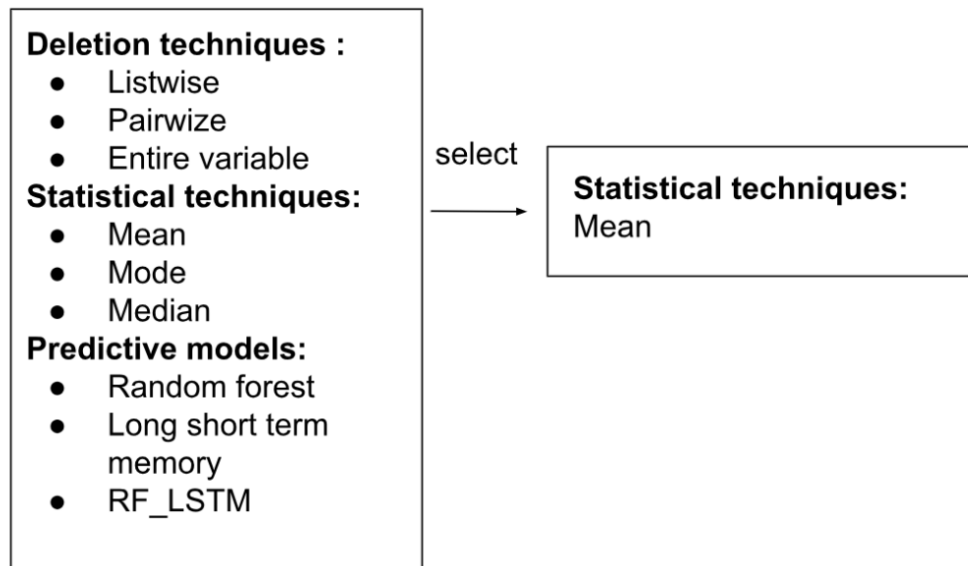


Figure 4.7: Example of suggested methods for handling missing data

In example 4.7, a statistical technique is deployed to calculate the missing data. Indeed, the mean parameter is used as a relevant value in this example. At this stage, the system allows the following techniques for handling missing data.

#### A) Handling missing data using statistical techniques

This part explains the approach to dealing with missing data using methods of imputing the mean, median, and mode. As we show in figure 4.8.

- The first step involves identifying the data types of each column in the dataset. Columns are categorized as: categorical data, numerical data, and other types (which are converted to categorical data).
- Categorical data needs to be encoded to facilitate imputation. Label encoding is used to convert categorical values into numerical values, which allows for the calculation of statistical measures such as the mean and median.
- For each column with missing values:
  - Numerical data:** The missing data are replaced with the mean, median, or mode of the column.
  - Categorical data:** The encoded values are used to calculate the mean or median, which are then decoded back to the original categories. The mode is directly used as it is already a category.

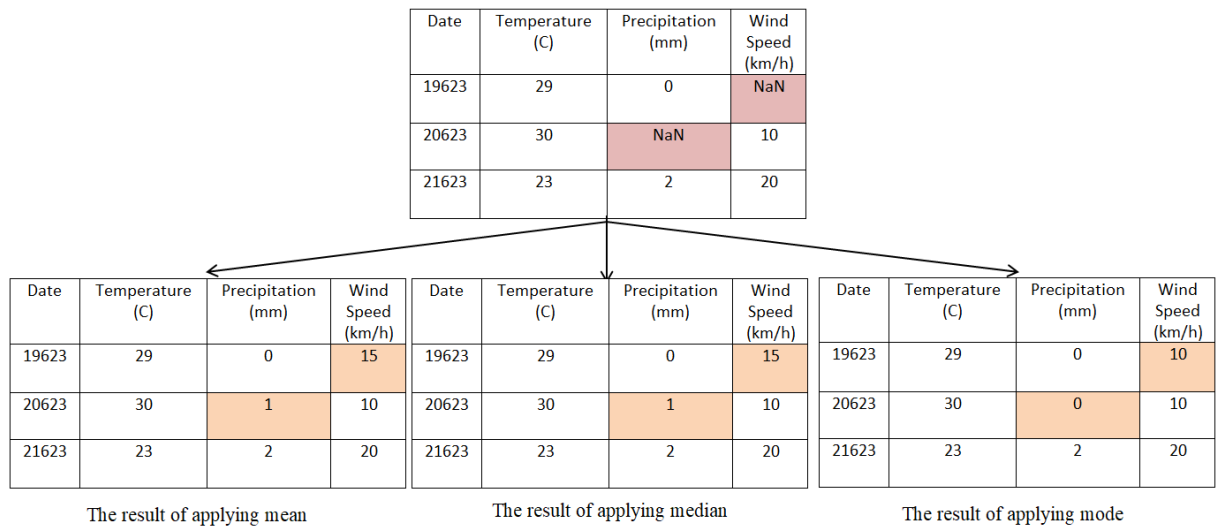


Figure 4.8: An example of the operation of statistical techniques

### B) Handling missing data using deletion techniques

This function refers three deletion methods used: list deletion, binary deletion, and entire variable deletion. As we show in Figure 4.9.

- **Listwise deletion:** We scan the dataset to identify any missing values in the records (rows). Then, we exclude any row that contains one or more missing values. Finally, we ensure that the resulting dataset does not contain any missing values.
- **Pairwise deletion:** We leave users the flexibility to choose specific columns for binary deletion, indicating which columns to analyze and check for missing values. Only these selected columns undergo scrutiny for missing data. Subsequently, for these chosen columns, we exclusively remove rows containing any missing values. Once cleaned, the dataset is either presented to the user for review or seamlessly utilized in further analytical procedures as required.
- **Delete the entire variable:** We conducted a comprehensive scan of the dataset to detect columns with missing values. If any column is found to have one or more missing values, it is automatically excluded from the dataset. This rigorous process guarantees that the resulting dataset is completely devoid of any columns containing missing values.

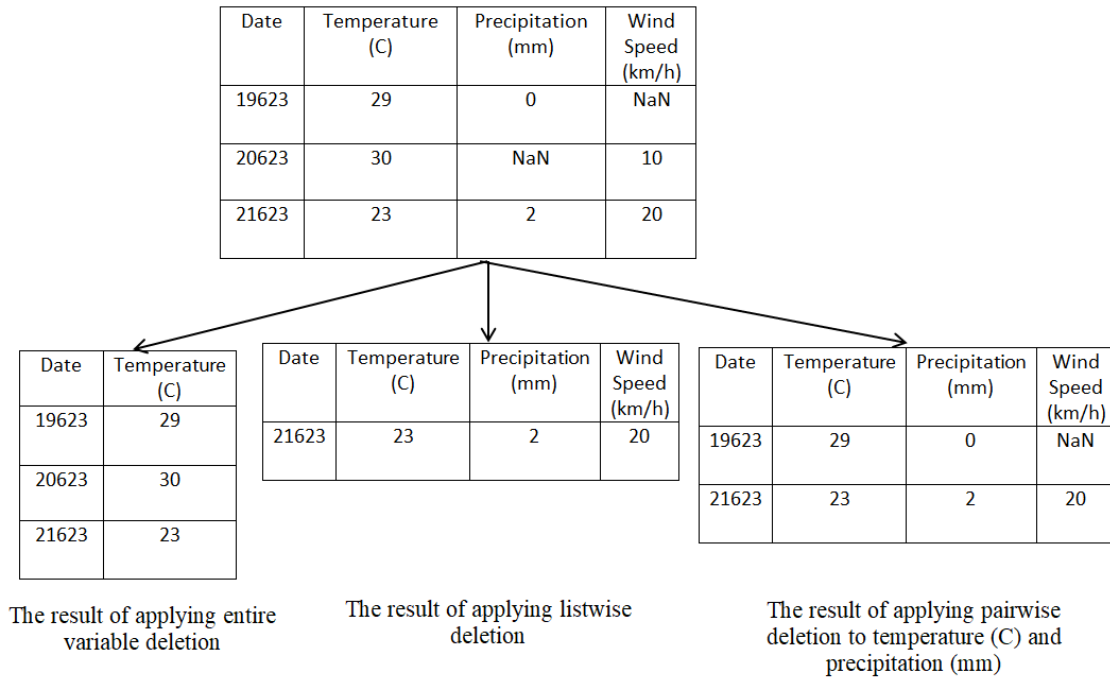


Figure 4.9: An example of the operation of deletion techniques

### C) Handling missing data using An enhanced techniques

AI models can consistently contribute to addressing the challenge of missing data. In what follows, we explore the usefulness of three models: random forest, LSTM, and hybrid model (RF\_LSTM). By leveraging these techniques, we aim to address the pervasive challenge of missing data.

A data preprocessing stage is initially performed in both methods (random forest and LSTM), where the dataset is first loaded, followed by extracting the year, month, and day from the date column, and then randomly inserting missing values into the dataset, adhering to a specified missing percentage. These preparatory procedures lay the foundation for subsequent analysis and model training.

#### a) Random forest (RF) model

The framework implements Random Forest model for managing missing data in the weather dataset, the following steps were undertaken.

**Model setup:** In the model setup phase, parameters for the Random Forest model are configured, laying the foundation for subsequent hyperparameter tuning through **Grid search** which is a hyperparameter tuning technique that systematically tests combinations of hyperparameter values to find the best model performance. Specifically, the following parameters are configured:

- **Random state:** This parameter ensures reproducibility by seeding the random number generator. It is set to 42 providing consistency in model initialization and

training.

Additionally, a parameter Grid is defined to specify the hyperparameters to be tuned through Grid Search, the Grid consists of the following hyperparameters:

- **Number of estimators:** Grid search is performed over the values [100, 200, 300], allowing the algorithm to determine the optimal number of decision trees in the set.
- **Maximum depth:** Grid search explores different values for the maximum depth of each decision tree, including none, 10, 20, and 30. This parameter controls the maximum depth of the individual decision trees in the set.
- **Minimum samples split:** Grid search considers various values for the minimum number of samples required to split an internal node, such as 2, 5, and 10. This parameter regulates the minimum number of samples required to split a node during tree building.
- **Minimum samples leaf:** Grid search evaluates different values for the minimum number of samples required to be at a leaf node, including 1, 2, and 4. This parameter sets the minimum number of samples required to be at a leaf node.

**Model training:** In the training phase, each feature with missing values undergoes Grid Search to determine the optimal Random Forest model. This iterative process involves dividing the dataset into training and test sets, performing hyperparameter tuning using Grid Search and selecting the best model based on the search results.

**Testing and saving the trained model:** Following the training phase, the trained Random forest model is tested to impute missing values contained the test dataset. Additionally, the best performing Random Forest model is serialized and saved, ensuring accessibility and preservation for future use.

#### **b) Long short term memory (LSTM) model**

In addition to the random forest method, the framework implements the LSTM model for imputing missing data into the dataset. The following steps were undertaken to achieve this method:

**LSTM model setup:** For setting up the LSTM model for imputing missing data, parameters including input size and output size are configured according to the dataset's dimensions and hidden size, which is equal to 64 layers. The model architecture comprised an LSTM layer followed by a linear layer for prediction.

**Training the LSTM model:** During the training phase, the LSTM model is trained using a mean squared error loss function and optimized with the Adam optimizer (*the Adam optimizer is an adaptive learning rate optimization algorithm*), where the training

loop iterates for a total of 50 epochs and processes batches of data iteratively to update the model parameters and minimize the loss function, enhancing the model's ability to impute missing values accurately.

**Testing and saving the trained model:** The trained LSTM model is tested to impute missing values on the test dataset. Subsequently, the imputed data was saved for further analysis, and then, post-training, the model parameters were saved to a file to ensure accessibility and preservation for future use.

### c) Hybrid (RF\_LSTM) model

Our important contribution is the combination of the two previous methods into a unique hybrid model. Indeed, the framework implements a hybrid imputation approach, combining the strengths of Random Forest (RF) and Long Short-Term Memory (LSTM) models to effectively handle missing data in datasets (*see figure 4.10*). Such a technique involves the following steps:

**Initial data processing:** The first phase of the hybrid imputation approach involves initial data processing. This begins with the identification of missing values within the dataset through the utilization of a boolean mask, effectively pinpointing the locations of missing entries. Missing values are then imputed using mean imputation, where the mean value of the respective column serves as the replacement for missing entries.

#### **Model integration and imputation:**

- The trained LSTM and RF models are loaded to enhance the imputation process, as values are predicted independently. The process is repeated through each feature with missing values.
- Integrate predictions from both models through a weighted average approach to capture the strengths of each model.
- Inserting imputed values derived from this hybrid method back into the data set effectively fills in gaps with more accurate estimates.

**Result analysis:** In the final stage, the processed data set, now complete with calculated values, is returned ready for subsequent analytical tasks, ensuring that data integrity is maintained and enhancing the reliability of any final analyses.

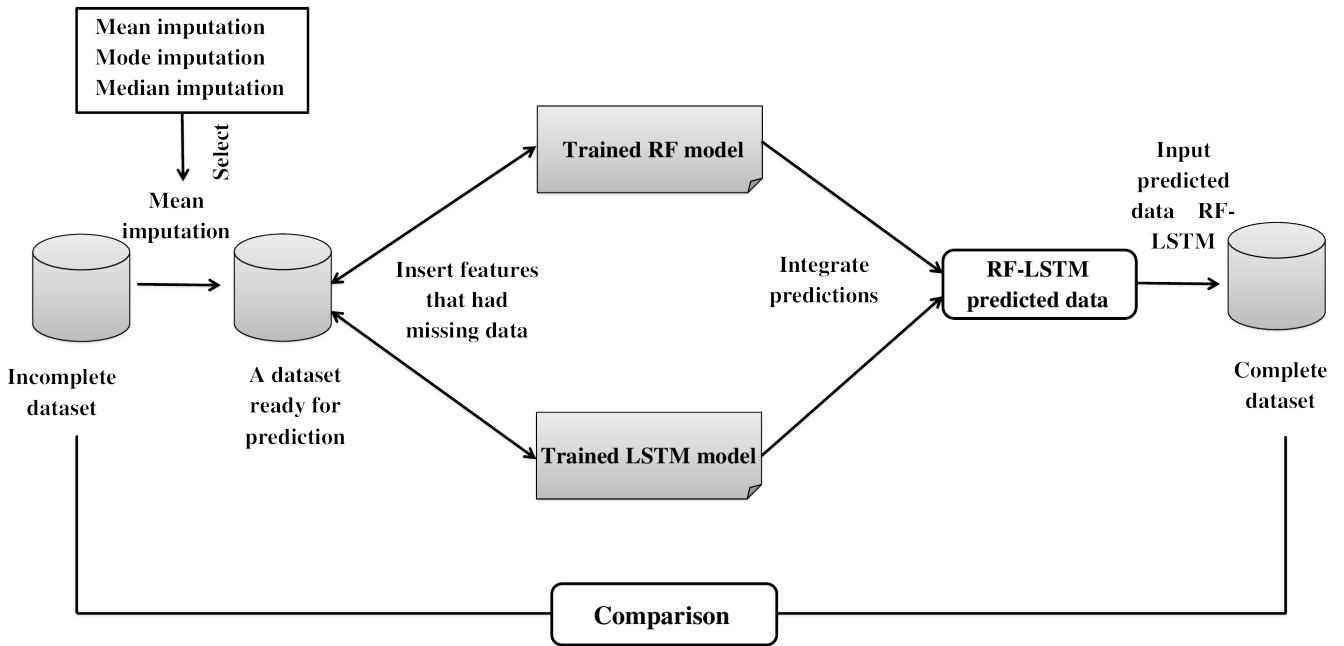


Figure 4.10: Hybrid (RF-LSTM) model mechanism

Once the suitable method is chosen, the corresponding algorithm is deployed to estimate the identified set of data to be processed. After missing data calculation, the resulting values are integrated, and the dataset is treated as data in its own right, like the original data (see figure 4.11).

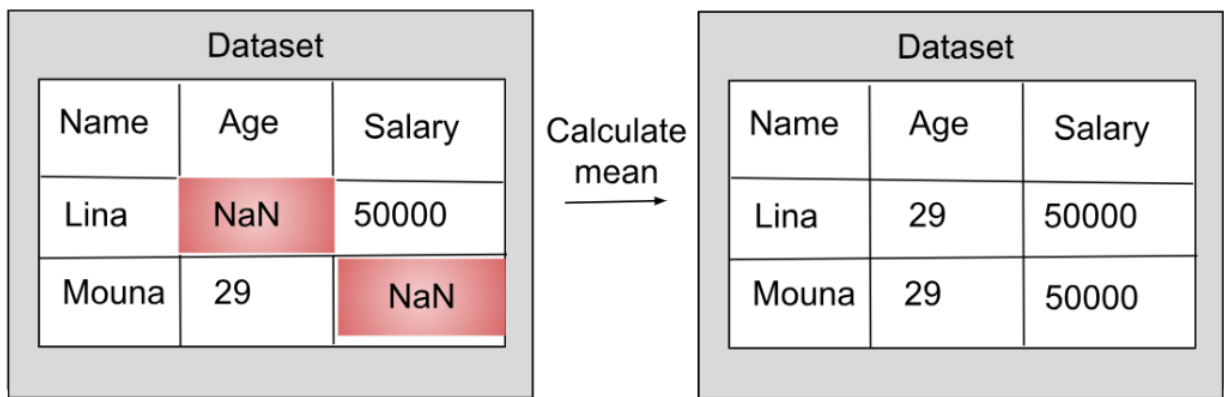


Figure 4.11: Example of missing data imputation

In the example 4.11, after applying the imputation of missing data, we have:

- **Input:** Incomplete dataset.
- Applying the imputation of missing data allows calculating the mean age from the available data. Replace the missing age with the calculated mean.
- **Output:** Complete dataset with calculated values.

### 4.7.5 Display calculated data

The complete processed data set is displayed, with the referenced data meticulously highlighted for easy identification (*see figure 4.12*).

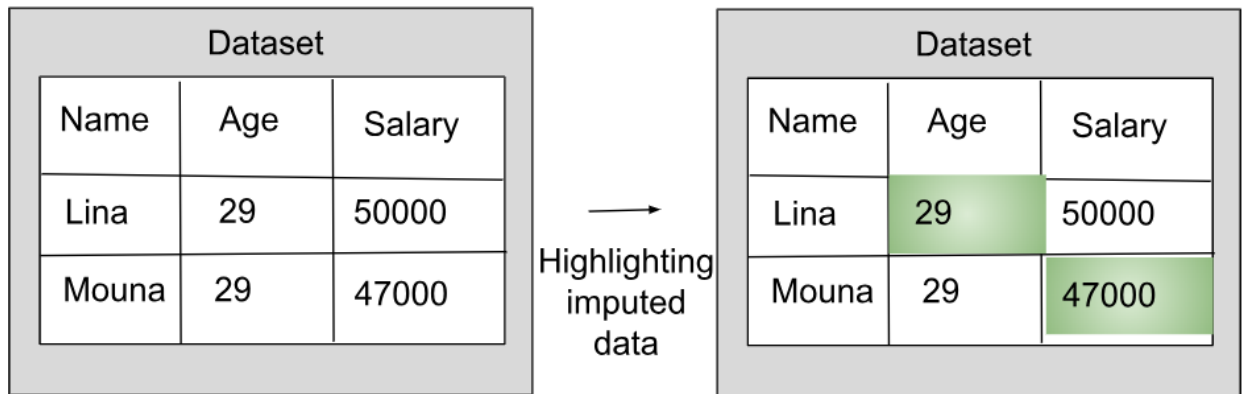


Figure 4.12: Example of display calculated data

In the example 4.12, after applying display calculated data, we find:

- **Input:** Complete dataset.
- **Applying display calculated data gives:** Data 29 of the age and 47,000 of the salary features are highlighted in green.
- **Output:** Values imputed to the dataset.

### 4.7.6 Statistics of the output dataset

This functionality allows for the assessment of the quality of the output dataset. It provides statistical parameters related to the provided dataset, such as the missing rate, the number of species and traits involved, and more. (*see figure 4.6*).

### 4.7.7 Handling a permanent dataset

The final step of the system consists of efficiently storing the newly completed dataset in a persistent storage system while ensuring data integrity and accessibility.

To gain a deeper understanding of the deployment and the advantages of the framework, in what follows, we illustrate its usage in a real-world scenario.

## 4.8 Usage scenario

In this section, we'll show a simple use-case scenario of our framework, focusing on handling missing data effectively.

1. **Data upload:** The user begins by introducing their dataset stored in an **CSV** file into the framework (*see table 4.3*). The progressive steps are discussed and explained.

Table 4.3: Incomplete dataset CSV

Day	Temperature (C)	Humidity (%)	Wind Speed (km/h)	Precipitation (mm)	Cloud Cover (%)	Visibility (km)	Air Quality Index
1	20	65	15	0	NaN	10	40
2	18	70	10	0.5	40	8	45
3	22	NaN	20	0	20	12	35
4	25	55	18	0	NaN	15	NaN
5	23	68	NaN	1	50	7	50
6	19	72	8	0.2	60	10	55
7	21	63	16	0	25	11	38
8	NaN	58	14	0	15	13	32
9	20	69	10	0.3	45	9	42
10	22	66	12	0	35	10	48
11	26	57	18	0	5	16	28
12	18	NaN	6	1.2	70	5	60
13	23	62	14	NaN	20	10	37
14	25	59	16	0	10	14	31
15	21	67	11	0.1	NaN	8	52

2. **Missing values detection and statistics:** The framework automatically detects any missing values in the dataset and provides relevant statistics, such as the number of missing values in each column (*see table 4.13*).

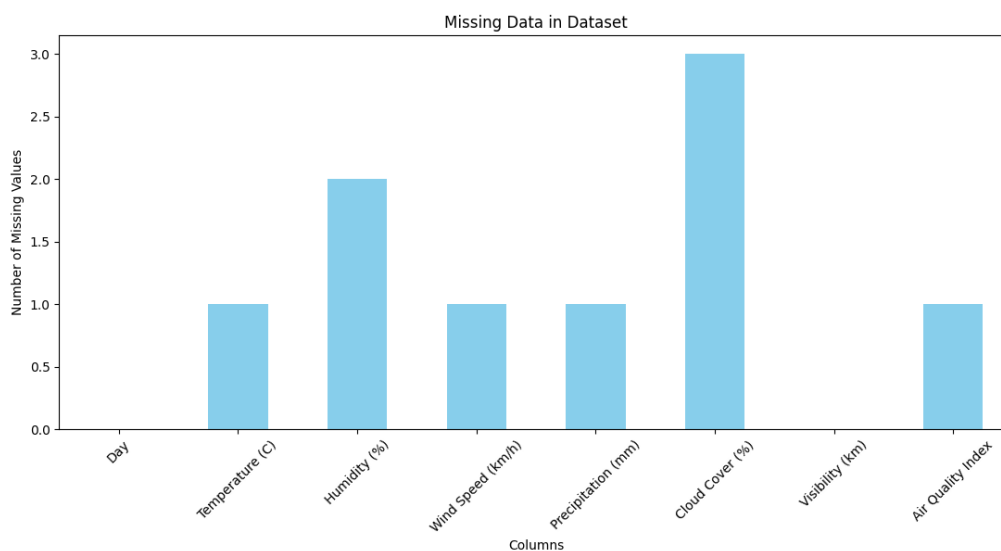


Figure 4.13: Number of missing values in each column



3. **Updated dataset and statistics:** Once the user chooses the appropriate method for calculating missing data, the system updates the dataset with calculated values using the mode method, and then it provides new statistical insights based on the adjusted data (see table 4.4).

Table 4.4: Complete dataset by mode method

Day	Temperature (C)	Humidity (%)	Wind Speed (km/h)	Precipitation (mm)	Cloud Cover (%)	Visibility (km)	Air Quality Index
1	20	65	15	0	<b>20</b>	10	40
2	18	70	10	0.5	40	8	45
3	22	<b>65</b>	20	0	20	12	35
4	25	55	18	0	<b>20</b>	15	<b>40</b>
5	23	68	<b>10</b>	1	50	7	50
6	19	72	8	0.2	60	10	55
7	21	63	16	0	25	11	38
8	<b>20</b>	58	14	0	15	13	32
9	20	69	10	0.3	45	9	42
10	22	66	12	0	35	10	48
11	26	57	18	0	5	16	28
12	18	<b>65</b>	6	1.2	70	5	60
13	23	62	14	<b>0</b>	20	10	37
14	25	59	16	0	10	14	31
15	21	67	11	0.1	<b>20</b>	8	52

4. **Exploring model predictive imputation:** To explore other imputation methods, the user selects the suitable predictive model among a list of proposed ones (see figure 4.14).

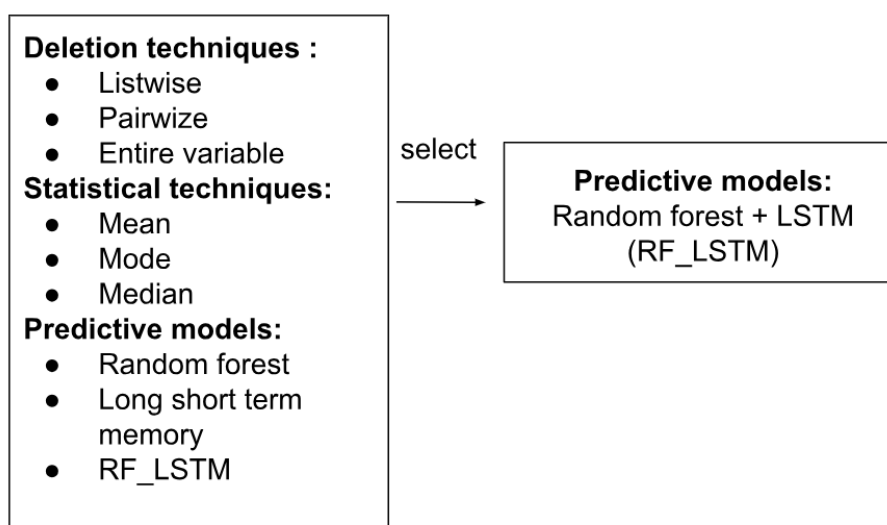


Figure 4.14: Selecting predictive model for imputation

5. **Dataset calculation:** The system applies the selected predictive imputation model to fill in missing values. As a result, new values for the missing data are calculated and stored in the appropriate variables. This creates a new dataset that reflects the calculated values (*see table 4.5*).

Table 4.5: Dataset imputation with predictive models

Day	Temperature (C)	Humidity (%)	Wind Speed (km/h)	Precipitation (mm)	Cloud Cover (%)	Visibility (km)	Air Quality Index
1	20	65	15	0	<b>30</b>	10	40
2	18	70	10	0.5	40	8	45
3	22	<b>60</b>	20	0	20	12	35
4	25	55	18	0	<b>10</b>	15	<b>30</b>
5	23	68	<b>12</b>	1	50	7	50
6	19	72	8	0.2	60	6	55
7	21	63	16	0	25	11	38
8	<b>24</b>	58	14	0	15	13	32
9	20	69	10	0.3	45	9	42
10	22	66	12	0	35	10	48
11	26	57	18	0	5	16	28
12	18	<b>75</b>	6	1.2	70	5	60
13	23	62	14	<b>0</b>	20	10	37
14	25	59	16	0	10	14	31
15	21	67	11	0.1	<b>55</b>	8	52

6. **Comparative evaluation:** Now, the user wishes to compare the effectiveness of mode imputation versus model predictive imputation. The system facilitates this task by presenting analyses of the datasets, highlighting differences in statistical measures and data distribution.
7. **Decision making on final data:** Based on the insights provided by the comparative evaluation, the user can make informed decisions about the imputation method that best matches the characteristics of the data and the objectives of the analysis.

## 4.9 Conclusion

In this chapter, we have conceived a comprehensive framework that tackles the issue of missing data. The proposed system incorporates various methods for assessing missing data. It allows the user to select statistical methods (mean, mode, median) for calculating missing data, then handles the appropriate predictive models (*Random Forest, LSTM, RF\_LSTM*) to ensure that the calculated values are accurate and credible, in addition to deletion methods (*listwise, pairwise, entire variable*). Moreover, it provides a deep insight into the complexities surrounding the missing data phenomenon, including its causes, patterns, and implications for analysis.

Chapter **5**

Implementation and experiments of the  
solution

## 5.1 Introduction

To show the feasibility of our conceived framework, we have implemented it in a software tool that integrates the functionalities useful for handling missing data by deploying various techniques.

This chapter exposes the development environment and software tools and their features, and then it discusses some experimental results.

## 5.2 Hardware environmental development

The hardware development environment refers to all the material and operating system resources used for the implementation of our system. Its description is shown in the table (*see table 5.1*):

Table 5.1: Characteristics of the used hardware

<b>Model</b>	<b>Part Used Laptop</b>
Modèle	PC portable
Processeur	Intel(R) Core(TM) i5-8365U CPU @ 1.60GHz 1.90 GHz
RAM	16,0 GB
System type	64-bit windows, x64-based processor

Before delving into the intricacies of entering elements into development tools, it is crucial to establish a foundational understanding of the tools themselves. To develop our system, we have used the following software environment.

## 5.3 The software environment

To develop our system for managing missing data, we have used and benefited from the following three tools.

### 5.3.1 PyCharm

Our system is crafted using the PyCharm Community Edition 2022.3.1 development environment, where PyCharm is an Integrated Development Environment developed by JetBrains and offers a wide array of features. PyCharm Community Edition operates under the Apache 2.0 License an open source license permitting free utilization of the software. The user interface of PyCharm Community Edition is designed to furnish developers with a comprehensive set of tools and make Python programming more accessible. It boasts a robust and feature-rich code editor. PyCharm seamlessly integrates with version control tools, facilitating collaborative work on software development projects [69].



Figure 5.1: Pycharm community 2022-03

The language used to develop our software application is Python, as explained in the section below.

### 5.3.2 Python language

Python was created by Guido van Rossum in 1991. The initial objective was to design a programming language that is simple, readable, and versatile, suitable for both beginners and experts alike. Python stands out for its clear and concise syntax, which promotes code readability and developer productivity. Python is widely used across various domains, from web development to data analysis and artificial intelligence, where its popularity stems in part from its ease of learning and its large developer community, which contributes to enriching its ecosystem with tools and libraries [70]. Here are some key features of Python:

- Its simple and clear syntax, which promotes code readability, reduces the amount of code needed to accomplish a given task.
- Python's dynamic interpretation enables execution of the code without preliminary compilation.
- Its portability across multiple platforms, including Windows, macOS, and Linux, is achievable.
- Its wide variety of libraries and frameworks makes it easier to develop applications in various domains.

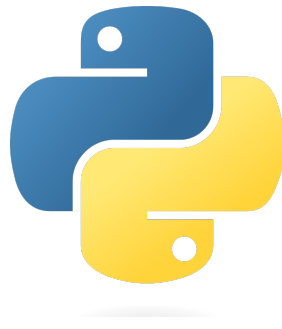


Figure 5.2: Python logo [18]

In what follows, the considered libraries are presented.

### 5.3.3 Python libraries

The Python ecosystem includes a large spectrum of libraries offering various functions.

1. **Torch:** The Torch library in Python, known as PyTorch, is a popular open-source machine learning library developed by Facebook's AI Research Lab. PyTorch provides a flexible and efficient platform for deep learning and tensor computation. It is widely used for developing neural networks due to its dynamic computation graph, which allows for more intuitive model building and debugging [71].
2. **Scikit-learn:** Scikit-learn, or Sklearn, is the most reliable and practical Python Machine Learning (ML) library. Through a Python consistency interface, it offers a range of effective tools for statistical modeling and machine learning, including regression, clustering, classification, and dimensionality reduction. This library is based on NumPy, SciPy, and Matplotlib and is mostly developed in Python [72].
3. **PyQt6:** PyQt is a derivative of the well-known Qt cross-platform GUI framework. It is the outcome of fusing the potent Qt library with the flexible Python language. To put it more technically, PyQt6 is a library wrapper for Qt6. PyQt6, the most recent version, allows us to construct sleek, portable, and contemporary graphical user interfaces for our Python scripts [73].
4. **Pandas:** Pandas is an open-source Python library that stands as a powerhouse for data manipulation and analysis tasks. It also offers a comprehensive set of data structures and functions tailored for efficient operations on datasets, empowering users to perform a wide range of data manipulation and analysis tasks with ease [74].
5. **Matplotlib:** Matplotlib serves as a comprehensive Python library enabling the creation of static, animated, and interactive visualizations. Matplotlib simplifies the creation of basic visualizations while providing the flexibility to tackle complex visualization tasks with its intuitive interface [75].

6. **NumPy:** NumPy serves as an indispensable open-source Python library utilized across various scientific and engineering domains, where it stands as the universal standard for handling numerical data within the Python ecosystem. NumPy caters to a diverse user base and offers powerful tools and functionalities for efficient numerical data manipulation and analysis [76].
7. **Tkinter:** Tkinter, a Python library, enables the development of fundamental graphical user interface (GUI) applications and holds the distinction of being the most commonly utilized module for GUI applications within the Python programming environment [77].
8. **mysql.connector:** The mysql.connector library is a Python driver for MySQL databases. It allows Python programs to connect to MySQL database servers and execute SQL queries. This library provides an interface to interact with MySQL databases by handling tasks such as establishing connections, executing queries, fetching results, and managing transactions [78].

It is important to notice that the previously mentioned libraries are the most important used ones. In fact, other libraries are deployed to achieve complementary functions. Now, we present the developed system and its components and interfaces. The developed software tool is named Hybrid Handling Missing Data ([H2MD](#)).

## 5.4 System overview

We devote this section to presenting the interface and functions of our system, with special emphasis on the main modules. We start by presenting some interfaces and the main modules of the implemented system.

### 5.4.1 Interfaces

In this sub-section we provide an overview of the interfaces available in the software application. As shown in figure 5.3, the [H2MD](#) provides preparation screen that appears when we invoke it.

This multilingual support ensures that users can navigate and utilize the application comfortably in their preferred language, enhancing overall accessibility and user experience, as depicted in figure 5.4. The [H2MD](#) offers both an English version, as shown in figure 5.5, and an Arabic version, as shown in figure 5.6, to facilitate ease of use for a diverse range of users.

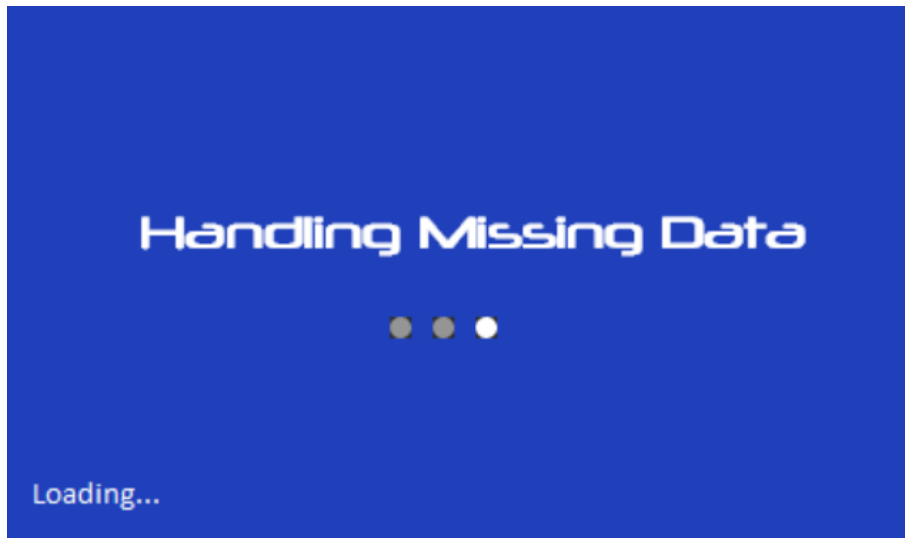


Figure 5.3: Splash screen



Figure 5.4: Home interface

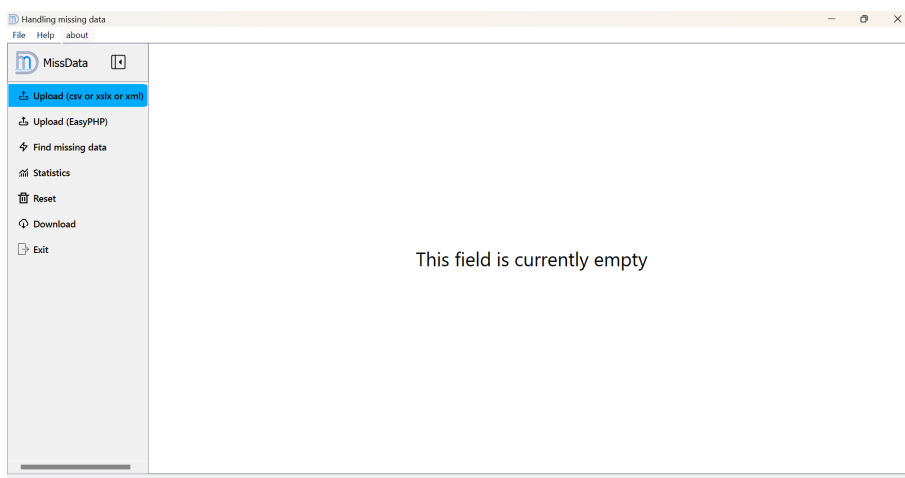


Figure 5.5: English main interface



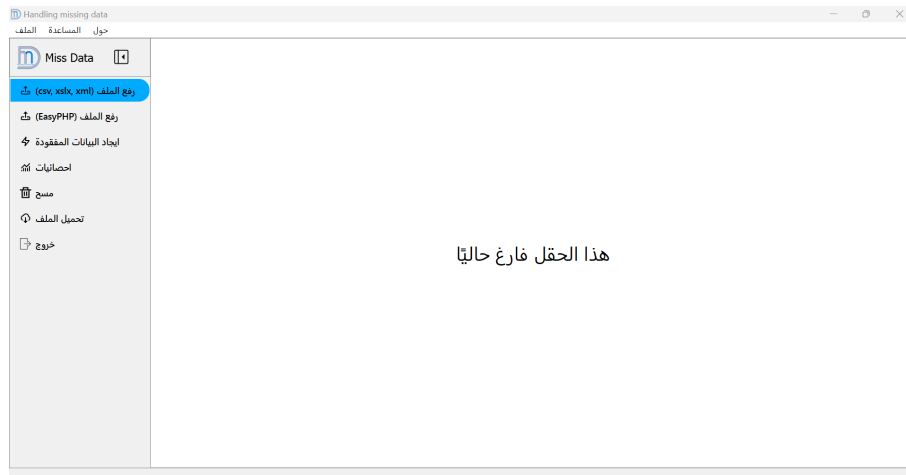


Figure 5.6: Arabic main interface

Now, we expose the main modules of the implemented system.

### 5.4.2 Main modules

The main modules of our software application **H2MD** are depicted in the following numbers (see figure 5.7)

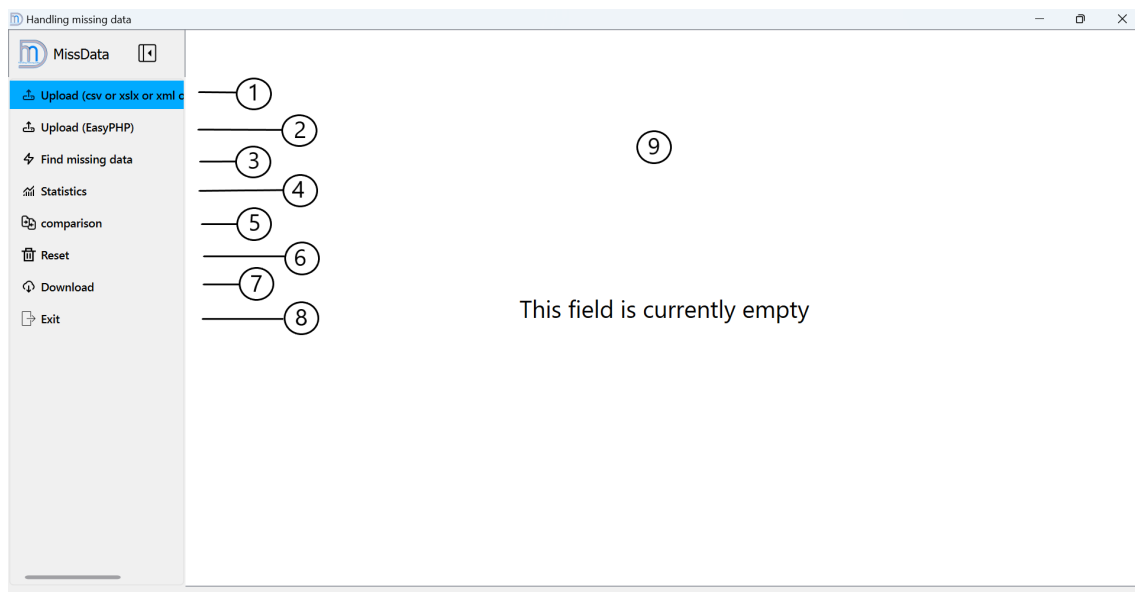


Figure 5.7: The main modules of H2MD

1. Upload datasets to be processed in various formats (**CSV**, **XLSX**, **XML**, **DOCX**, **PDF**).
2. Upload the dataset to be processed using EasyPHP.
3. Provide the user with the ability to process missing data in the uploaded datasets.

4. Display statistical indicators about the total data before and after imputing missing data.
5. Present a comparison of the methods used to handle missing data.
6. Allow the reset of all entries in the database and the results.
7. Provide the user with the ability to download the processed dataset.
8. A special part of the interface to display the results.

## 5.5 Exploration of H2MD functionalities

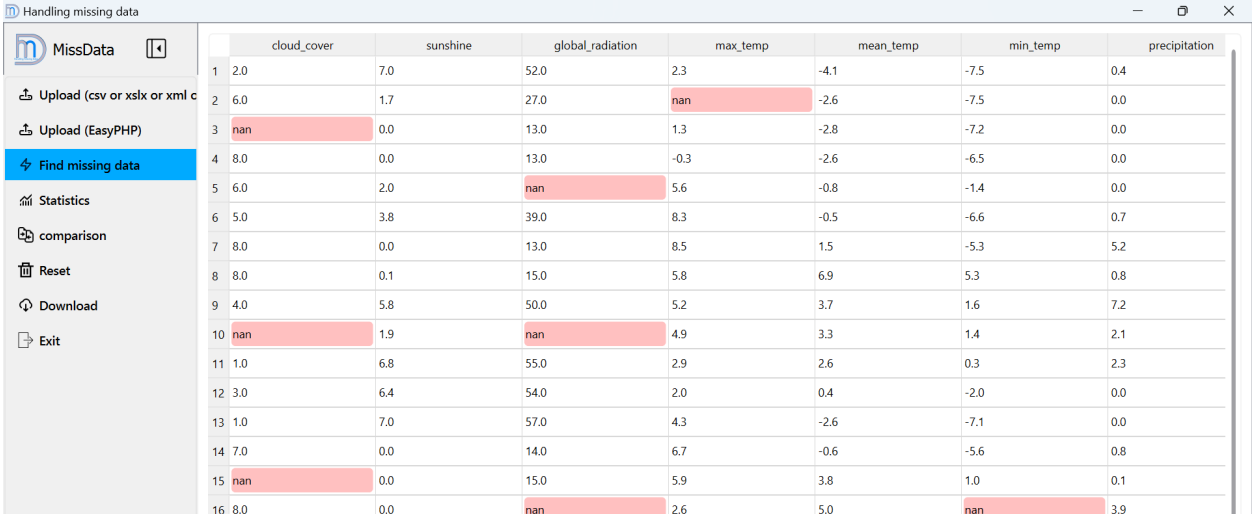
In this section, we will outline the fundamental principles of our system's functioning in English version mode. We will explain the various stages involved in processing missing data and their induced impacts on data completion and storage. This can be divided into two sections based on input types: from file format and from the EasyPHP dataset.

### 5.5.1 Handling missing data in various file formats

Our software application can upload different types of files, including [CSV](#), [XLSX](#), [XML](#), [DOCX](#), and [PDF](#), where each file type serves different purposes and may contain specific data formats.

1. To begin the process of handling missing data, we start by uploading the dataset using the "Upload" option. This step allows us to access the dataset to be processed, with missing data locations highlighted in red for easy identification.

As shown in figure 5.8, this visual indicator helps users quickly locate and address the missing values in the managed data.



The screenshot shows a web application window titled "Handling missing data". On the left is a sidebar menu with options: "MissData", "Upload (csv or xlsx or xml)", "Upload (EasyPHP)", "Find missing data" (highlighted in blue), "Statistics", "comparison", "Reset", "Download", and "Exit". The main area displays a table with 8 columns: "cloud\_cover", "sunshine", "global\_radiation", "max\_temp", "mean\_temp", "min\_temp", and "precipitation". The rows are numbered 1 to 16. Missing values are indicated by "nan" in red cells: row 2 (max\_temp), row 3 (cloud\_cover), row 5 (global\_radiation), row 10 (cloud\_cover and global\_radiation), row 15 (cloud\_cover), and row 16 (global\_radiation and min\_temp).

	cloud_cover	sunshine	global_radiation	max_temp	mean_temp	min_temp	precipitation
1	2.0	7.0	52.0	2.3	-4.1	-7.5	0.4
2	6.0	1.7	27.0	nan	-2.6	-7.5	0.0
3	nan	0.0	13.0	1.3	-2.8	-7.2	0.0
4	8.0	0.0	13.0	-0.3	-2.6	-6.5	0.0
5	6.0	2.0	nan	5.6	-0.8	-1.4	0.0
6	5.0	3.8	39.0	8.3	-0.5	-6.6	0.7
7	8.0	0.0	13.0	8.5	1.5	-5.3	5.2
8	8.0	0.1	15.0	5.8	6.9	5.3	0.8
9	4.0	5.8	50.0	5.2	3.7	1.6	7.2
10	nan	1.9	nan	4.9	3.3	1.4	2.1
11	1.0	6.8	55.0	2.9	2.6	0.3	2.3
12	3.0	6.4	54.0	2.0	0.4	-2.0	0.0
13	1.0	7.0	57.0	4.3	-2.6	-7.1	0.0
14	7.0	0.0	14.0	6.7	-0.6	-5.6	0.8
15	nan	0.0	15.0	5.9	3.8	1.0	0.1
16	8.0	0.0	nan	2.6	5.0	nan	3.9

Figure 5.8: Example of uploading various datasets formats

- The "Find missing data" option allows the user to select a processing method from among several available methods (*statistical techniques, deletion techniques, predictive techniques*) along with the features to be processed in the loaded dataset, for example choosing the hybrid RF\_LSTM method to apply it to all features except "Cloud\_cover", as shown in figure 5.9.

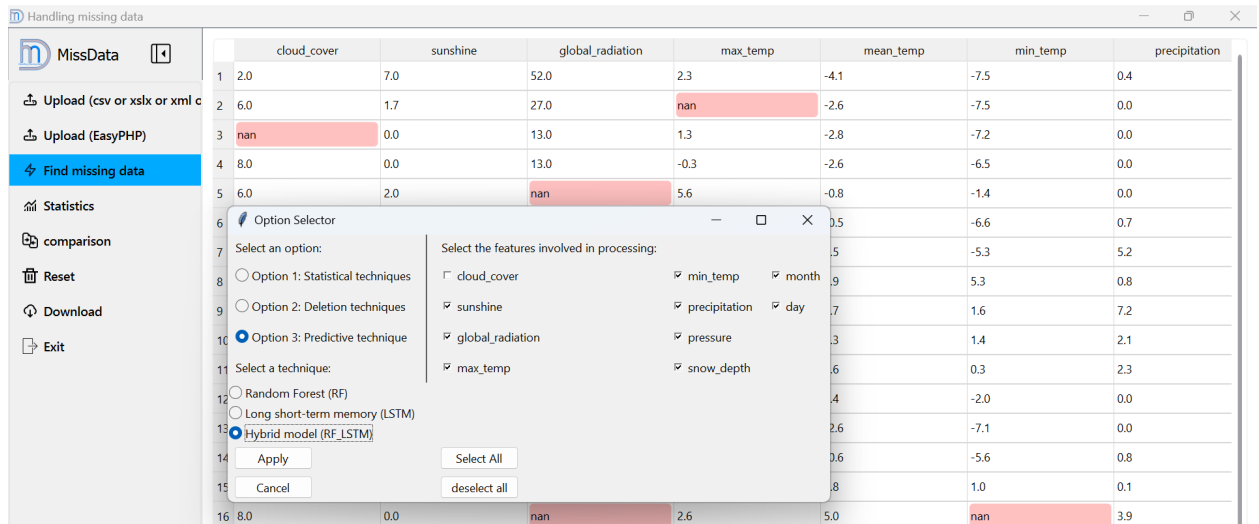


Figure 5.9: Illustration of the method ( RF\_LSTM ) and its features

- The dataset processed by the hybrid RF\_LSTM method, which takes features of both Random Forest and LSTM models to improve prediction accuracy, is shown with assigned data highlighted in green and excluded feature data highlighted in red. As shown in figure 5.10.

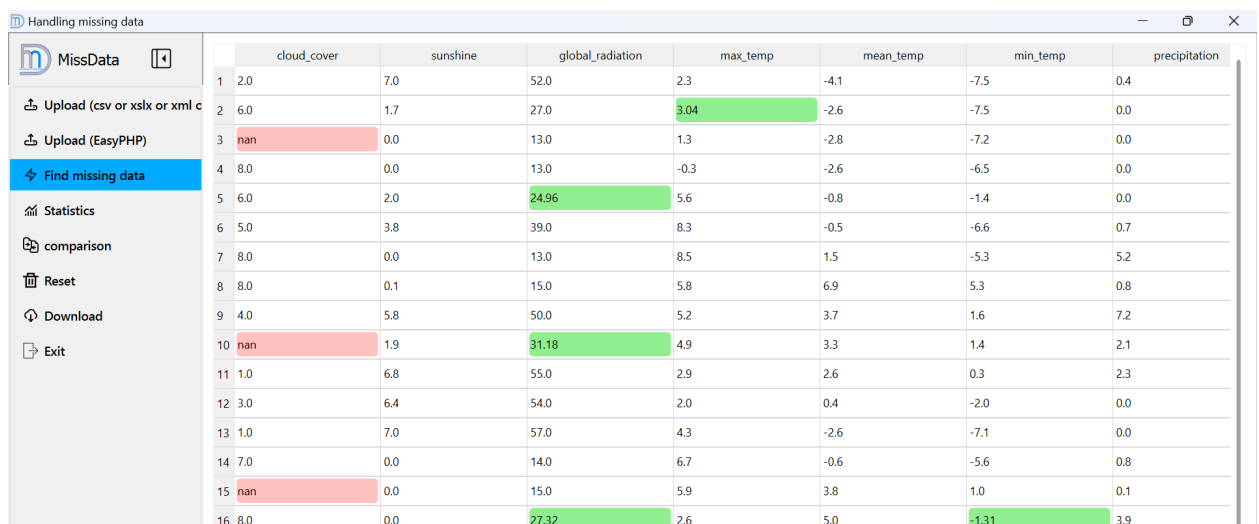


Figure 5.10: Example of processing missing data using RF\_LSTM

- The "Statistics" option displays statistics and graphs about missing data and data distribution before and after processing the data in the input and output datasets,

such as the number of missing data, the number of features, the name of the chosen method, etc. Which makes it easier for us to understand the impact of handling missing data 5.11.

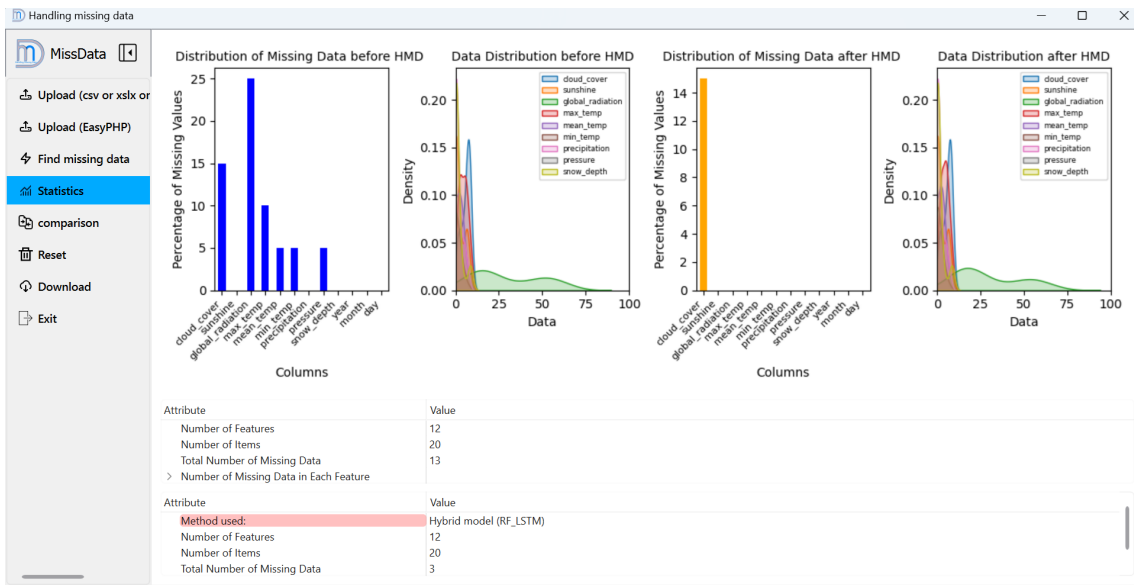


Figure 5.11: Statistics illustration of handling missing data

- Now, we reprocess the missing data in the dataset by employing listwise deletion using the "Find missing data" option. This method involves removing any rows with missing values, ensuring that our analysis is based on complete cases only. As illustrated in figure 5.12, this approach can simplify subsequent data analysis and improve the integrity of our results by excluding incomplete entries, though it should be used cautiously as it can lead to the loss of valuable information if many rows are deleted.

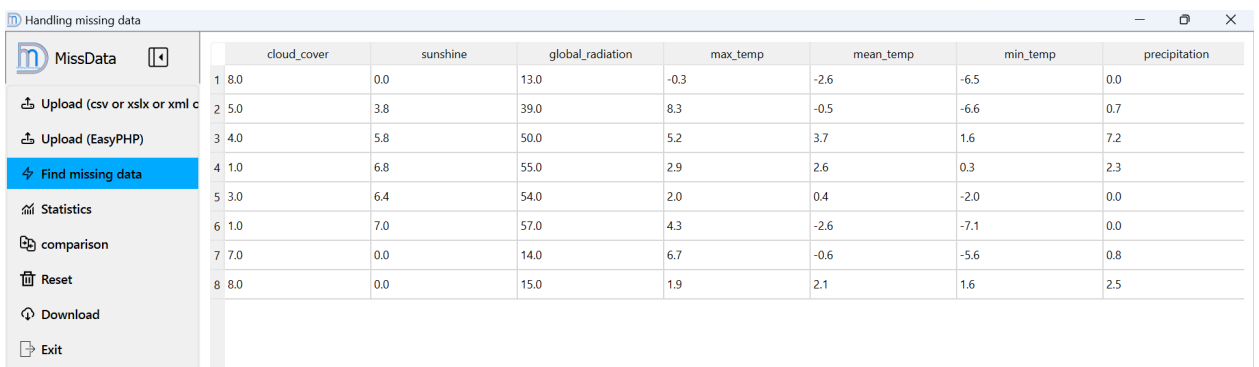


Figure 5.12: Example of processing data using Listwise deletion

6. The "Comparison" option evaluates the methods used based on a set of criteria, such as their execution time (*if applicable*), and the size of the processed dataset. This functionality helps in determining the most efficient and accurate method for handling missing data. As shown in figure 5.13, users can easily compare these metrics to make informed decisions on the best approach for their specific data processing needs.

Additionally, the tool provides visualizations that highlight the strengths and weaknesses of each method. Users can customize the comparison criteria to align with their specific objectives.

This tailored evaluation ensures that the selected method optimally addresses the unique challenges of their dataset.

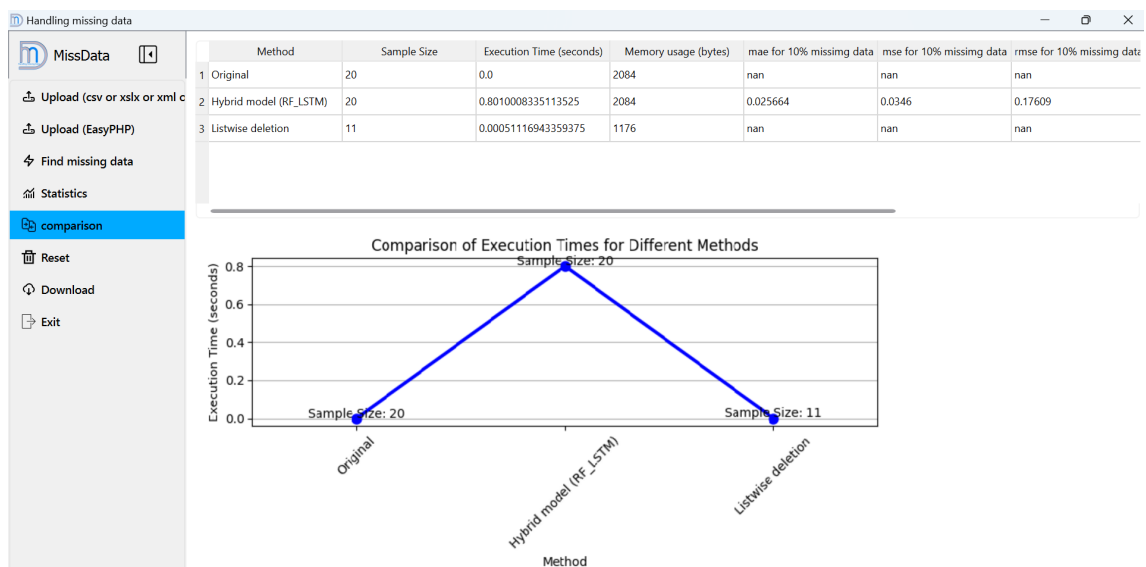


Figure 5.13: Comparison of the used methods

7. The "Download" option allows users to download the processed dataset in CSV format, while saving it under the name *filename.csv*.

This feature provides a convenient way to export the cleaned and processed data for further analysis or sharing. As shown in figure 5.14, this functionality ensures that users can easily obtain their dataset in a widely portable used format.

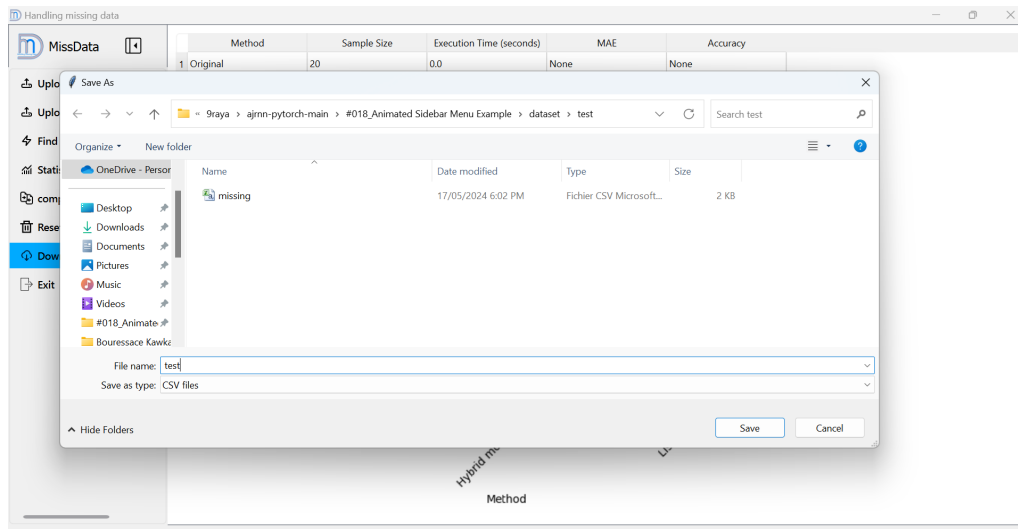


Figure 5.14: Illustration of downloading the processed dataset in csv format

### 5.5.2 Handling missing data with easyPHP

Our software application H2MD includes the ability to work on databases through easyPHP, enhancing functionality for seamless file uploads. Our software application H2MD includes.

1. Ensures that easy PHP is activated and that the considered dataset exists on adequate support, as depicted in figure 5.15.

cloud_cover	sunshine	global_radiation	max_temp	mean_temp	min_temp	precipitation	pressure	snow_depth	year	month	day
2	7	52	2.3	-4.1	-7.5	0.4	101900	9	1970	1	1
6	1.7	27	NULL	-2.6	-7.5	0	102050	8	1970	1	1
NULL	0	13	1.3	-2.8	-7.2	0	102050	4	1970	1	1
8	0	13	-0.3	-2.6	-6.5	0	100840	2	1970	1	1
6	2	NULL	5.6	-0.8	-1.4	0	102250	1	1970	1	1
5	3.8	39	8.3	-0.5	-6.6	0.7	102780	1	1970	1	1
8	0	13	8.5	1.5	-5.3	5.2	102520	0	1970	1	1
8	0.1	15	5.8	6.9	5.3	0.8	101170	0	1970	1	1
4	5.8	50	5.2	3.7	1.6	7.2	101170	0	1970	1	1
NULL	1.9	NULL	4.9	3.3	1.4	2.1	98700	0	1970	1	1
1	6.8	55	2.9	2.6	0.3	2.3	98960	0	1970	1	1
3	6.4	54	2	0.4	-2	0	100650	1	1970	1	1
1	7	57	4.3	-2.6	-7.1	0	102350	1	1970	1	1
7	0	14	6.7	-0.6	-5.6	0.8	102700	1	1970	1	1
NULL	0	15	5.9	3.8	1	0.1	102990	0	1970	1	1
8	0	NULL	2.6	5	NULL	3.9	103100	0	1970	1	1
8	0	15	1.9	2.1	1.6	2.5	102220	0	1970	1	1
8	0	NULL	3	0.8	-0.2	0.2	101860	0	1970	1	1
8	0	16	7.2	NULL	-1.4	5.2	100910	0	1970	1	1
7	0	NULL	NULL	3.1	-1	0	100920	0	1970	1	1

Figure 5.15: Example of managing a dataset in easyPHP format

2. The "upload EasyPHP" option allows you to select the desired database along with the dataset to be uploaded as shown in figure 5.16.

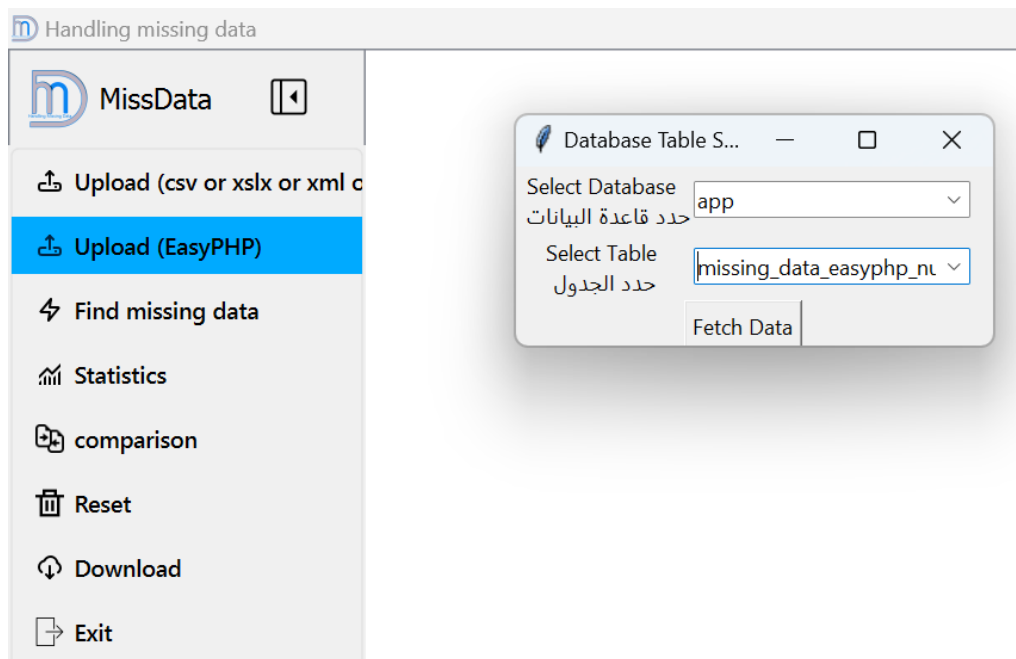


Figure 5.16: Example of how to select a database from a table or dataset

This step allows us to display the selected dataset, highlighting the locations of missing data in red for easy identification. As shown in figure 5.8, this ergonomic visual viewing helps users quickly identifying and addressing the missing values contained in their considered datasets.

3. The "Find missing data" option allows the user to select a processing method from among several available ones (*statistical techniques, deletion techniques, predictive techniques*) along with the features to be processed in the loaded dataset. For example, a specific use can choose the pairwise method from deletion methods in order to apply it to all features except "Cloud\_cover", as shown in figure 5.17.

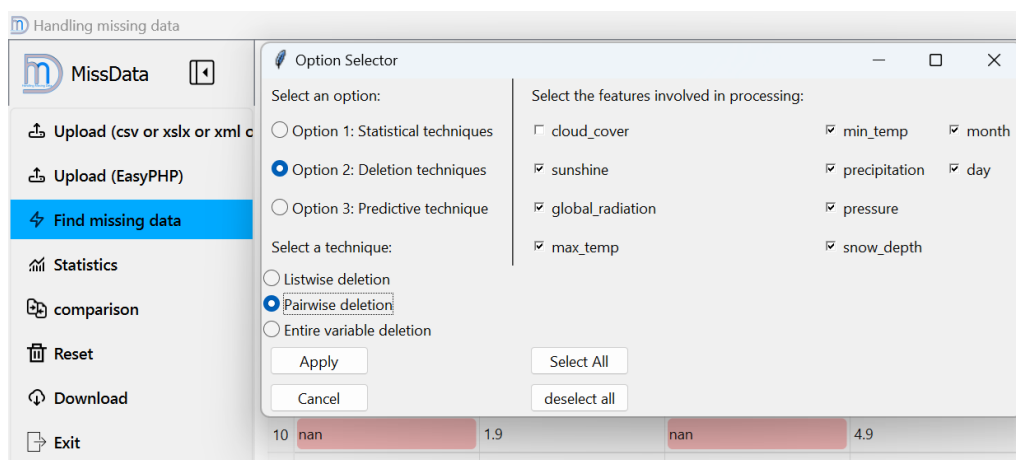


Figure 5.17: Illustration of selecting the pairwise method and features specification

As shown in figure 5.18, the user is allowed to select the features to be analyzed if the method Pairwise deletion is selected.

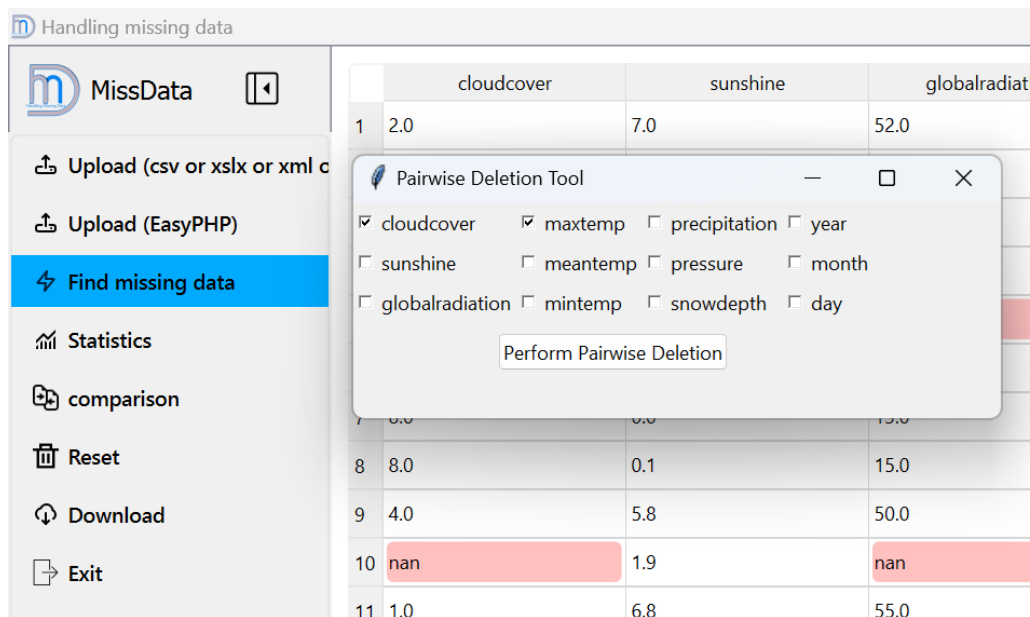


Figure 5.18: Example of selected features for data analysis

- The pairwise deletion method provides the user with the ability to remove only the specific data pairs where one or both values are missing, instead of removing entire rows or columns with missing values (*as in listwise deletion*). For example, the records containing missing data at the "cloudcover" and "maxtemp" levels are removed, while the remaining records are kept even if they contain missing data, as shown in figure 5.19.

The screenshot shows the MissData application interface with a processed dataset. The sidebar is the same as in Figure 5.18. The main area displays a data table with columns: cloudcover, sunshine, globalradiation, maxtemp, meantemp, mintemp, and precipitation. The table shows 15 rows of data. Some cells are highlighted in red, indicating missing values (nan): row 4 (cloudcover), row 12 (globalradiation), row 13 (maxtemp), row 14 (globalradiation), and row 15 (meantemp).

Figure 5.19: Example of a processed dataset using RF\_LSTM

- The "Statistics" option provides a display of statistics about the data set before and after handling missing data, as shown in figure 5.12. In addition, the "comparison" option provides the user with a view of the method that can be relied upon among the methods that he previously chose, as shown in figure 5.13. Our software application



allows the user to download the processed dataset in [CSV](#) format through the "Download" option, as shown in figure [5.14](#).

The following section presents the results obtained through careful experimentation and insightful analysis, highlighting their effectiveness and implications.

## 5.6 Results and analysis

In this section, we present the output of employing various techniques for handling missing data, encompassing statistical methods, deletion methods, and artificial intelligence approaches.

We evaluate the effectiveness of these techniques across different datasets, highlighting their strengths and limitations.

Before starting result analysis, it's important to identify relevant features to be used for our analysis.

**Evaluation indicators:** For evaluating the performance of the used techniques, several indicators are commonly used. These indicators help measure the accuracy and efficiency of the methods. Some of the main evaluation indicators are specified below.

- **Mean absolute error (MAE):** The average of the absolute differences between predicted values and actual values.

Measures the average magnitude of errors in predictions, without considering their direction [\[79\]](#).

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (5.1)$$

- **Mean squared error (MSE):** The average of the squared differences between predicted values and actual values.

Measures the average of the squares of the errors, penalizing larger errors more heavily [\[79\]](#).

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (5.2)$$

- **The root mean square error (RMSE):** RMSE is a standard way to measure the error of a model in predicting quantitative data.

It represents the square root of the average of the squared differences between the predicted and actual values [\[79\]](#).

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (5.3)$$

- **The Mean absolute percentage error (MAPE):** MAPE is a measure of prediction

accuracy of a forecasting method in statistics.

It expresses accuracy as a percentage, representing the average of the absolute percentage errors between the actual and predicted values [79].

$$\text{MAPE} = \frac{1}{n} \sum_{i=1}^n \left( \frac{|y_i - \hat{y}_i|}{|y_i|} \right) \times 100\% \quad (5.4)$$

$n$  : Represents the number of observations or data points for which the error is calculated.

$y_i$  : Denotes the actual observed values or ground truth values.

$\hat{y}_i$  : Denotes the predicted values generated by the model.

### 5.6.1 Results of statistical methods

In our implemented software application we use three popular metrics: mean, median and mode. We explained the steps in the previous chapter 4, where all 15342 records were used. The imputation result is shown in table 5.2 and figure 5.20.

Table 5.2: Statistical methods across different levels of missing data

Method	Missing Data (%)	MAPE (%)	MAE	MSE	RMSE
MEAN	10	56.1	831.06	94864.96	308.00
	40	77.2	790.89	96239.49	310.22
	60	82.6	752.23	90330.74	300.55
MEDIAN	10	62.8	710.16	85588.84	292.56
	40	71.6	741.60	88750.88	297.91
	60	82.8	737.46	86961.47	294.89
MODE	10	50.2	650.46	68493.25	261.71
	40	68.2	812.62	95788.33	309.50
	60	79.9	908.55	130353.01	361.04

The comparison of the three imputation methods (*mean*, *median*, *mode*) reveals distinct patterns in their performance across different levels of missing data. The mean method demonstrates consistent and moderate error metrics suggesting robustness in handling missing data. The median method shows comparable performance to mean but with slightly lower errors in certain scenarios. The mode method displays more varied outcomes with instances of both lower and higher errors, particularly evident in cases with higher levels of missing data.

These findings underscore the importance of carefully selecting an imputation method based on the specific characteristics and requirements of the dataset, ensuring reliable and accurate analyses.

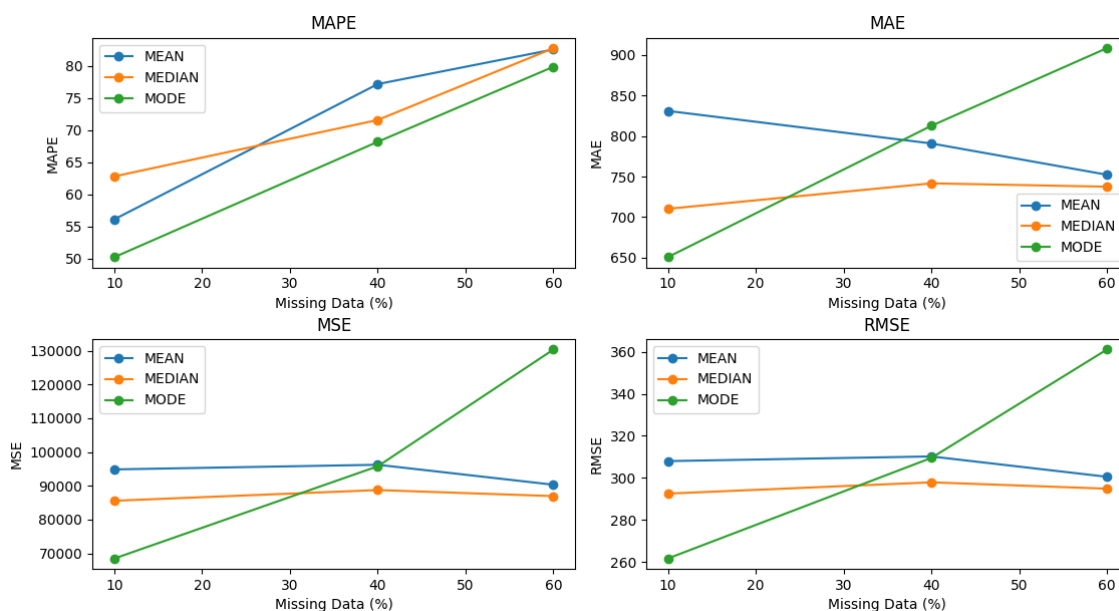


Figure 5.20: Graph of statistical methods across different levels of missing data

### 5.6.2 Results of deletion methods

In addition to the statistical results evaluation, H2MD software tool implements the listwise and pairwise methods, as well as entire variables. This aspect has been deeply explained in the previous chapter 4. As a consequence, the imputations results as depicted in table 5.3 and figure 5.20.

Table 5.3: Comparison of methods

Method	Missing Data (%)	Record Size	Number of Features	Number of records ignored
LISTWISE	0	15342	12	0
	10	13807	12	0
	40	1189	12	0
	60	5877	12	0
PAIRWISE	0	15342	12	0
	10	15342	12	988
	40	15342	12	4800
	60	15342	12	9912
ENTIRE VARIABLE	0	15342	12	0
	10	15342	11	0
	40	15342	11	0
	60	15342	4	0

The trends in the graph (*see figure 5.21*) illustrate the critical importance of choosing a particular method for handling missing data. It is clear that different deletion methods: listwise, pairwise, and entire variable provide varying results in terms of record size and number of features, especially in response to different levels of missing data.

These results emphasize the need to carefully consider the specific characteristics and requirements of a data set when choosing a deletion method to ensure the reliability and accuracy of subsequent analyses.



Figure 5.21: Graph of deletion methods across different levels of missing data

### 5.6.3 Results of artificial intelligence methods

In what follows, we present and analyze the results obtained by the three artificial intelligence methods.

#### Random forest (RF) model

In our system, we used Random Forest to handle missing data due to its robustness and ability to manage complex datasets with high dimensionality. This method not only improves the quality of the data but also enhances the overall performance of the model.

We explained the detailed steps of this imputation process in the previous chapter 4. The training results of the model, demonstrating its effectiveness in dealing with missing data, are presented in table 5.4 and figure 5.23.

Table 5.4: Random forest across different levels of missing data

Method	Missing Data (%)	MAPE (%)	MAE	MSE	RMSE
Random forest	10	2.60	0.0112	0.0115	0.1072
	40	7.65	0.01998	0.1081	0.319
	60	12.89	0.02333	0.2688	0.5184

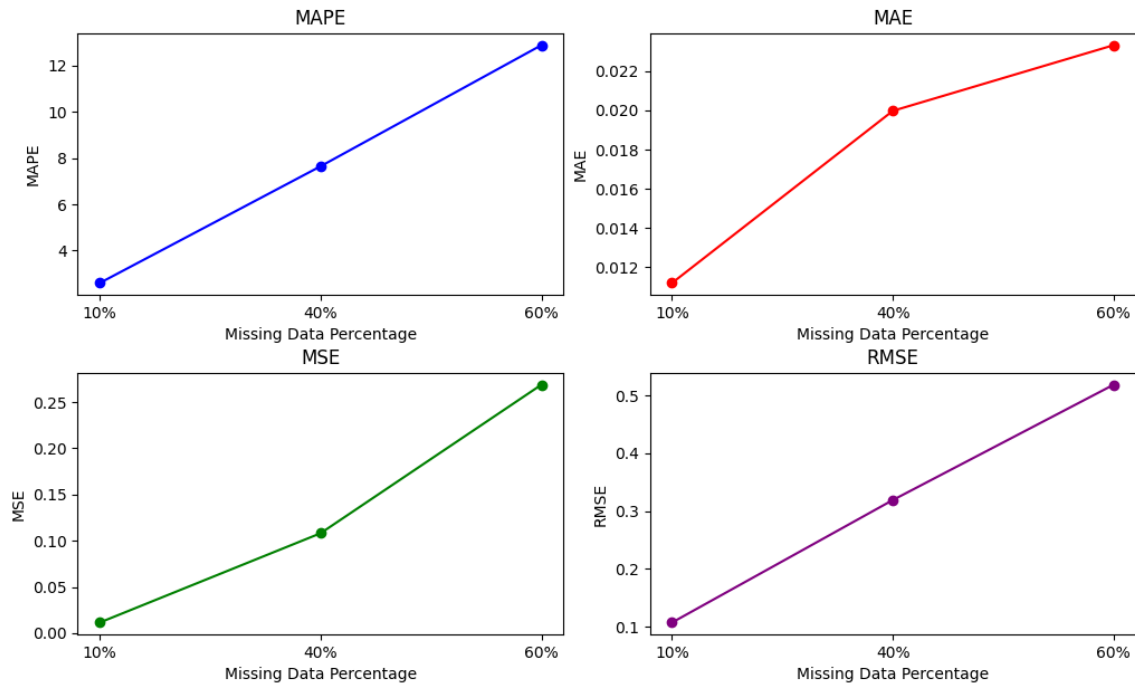


Figure 5.22: Graph of random forest model across different levels of missing data

The random forest model achieved a loss of 0.0142, indicating its effective reduction of discrepancies between expected and actual values. This suggests that the model's predictions are generally close to reality. Moreover, these findings demonstrate the model's high precision in handling missing data. In summary, the evaluation results affirm the efficacy and potential of the random forest model in addressing the presented problem, verifying its capability to deliver accurate forecasts, as illustrated in figure 5.23 of the loss curve.

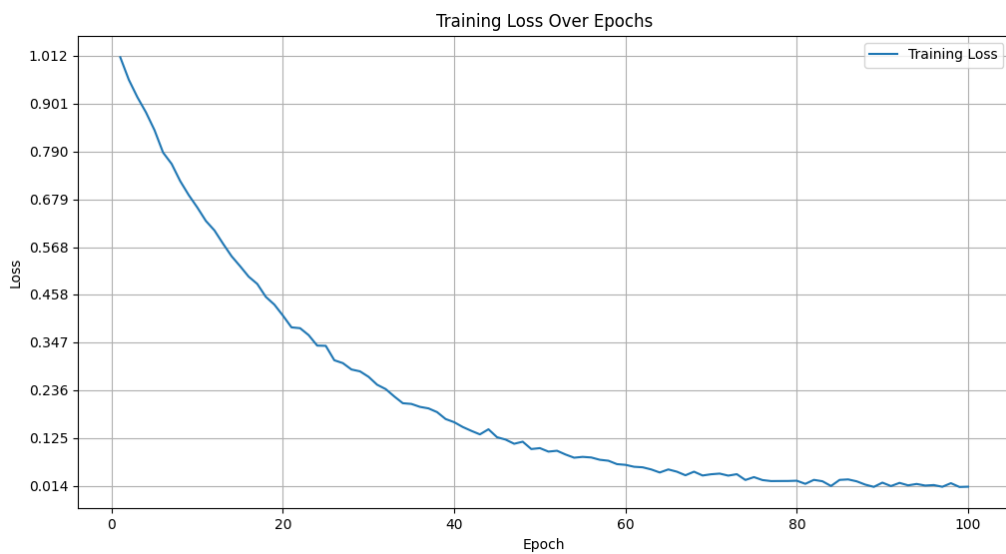


Figure 5.23: Training loss

### Long short-term memory (LSTM) model

In our system, we employed Long Short-Term Memory (LSTM) networks to handle missing data, capitalizing on their capability to capture temporal dependencies and patterns in sequential data.

The methodology for integrating LSTM networks into our system for missing data imputation is thoroughly discussed in the previous chapter 4. The performance results of our LSTM model, highlighting its effectiveness in addressing missing data issues, are displayed in table 5.5 and figure 5.24, where we find that the lower the percentage of missing data, the better the results.

Table 5.5: LSTM across different levels of missing data

Method	Missing Data (%)	MAPE (%)	MAE	MSE	RMSE
LSTM	10	3.39	0.075	0.1025	0.3201
	40	6.44	0.122	0.1555	0.3943
	60	14.67	0.181	0.2342	0.4839

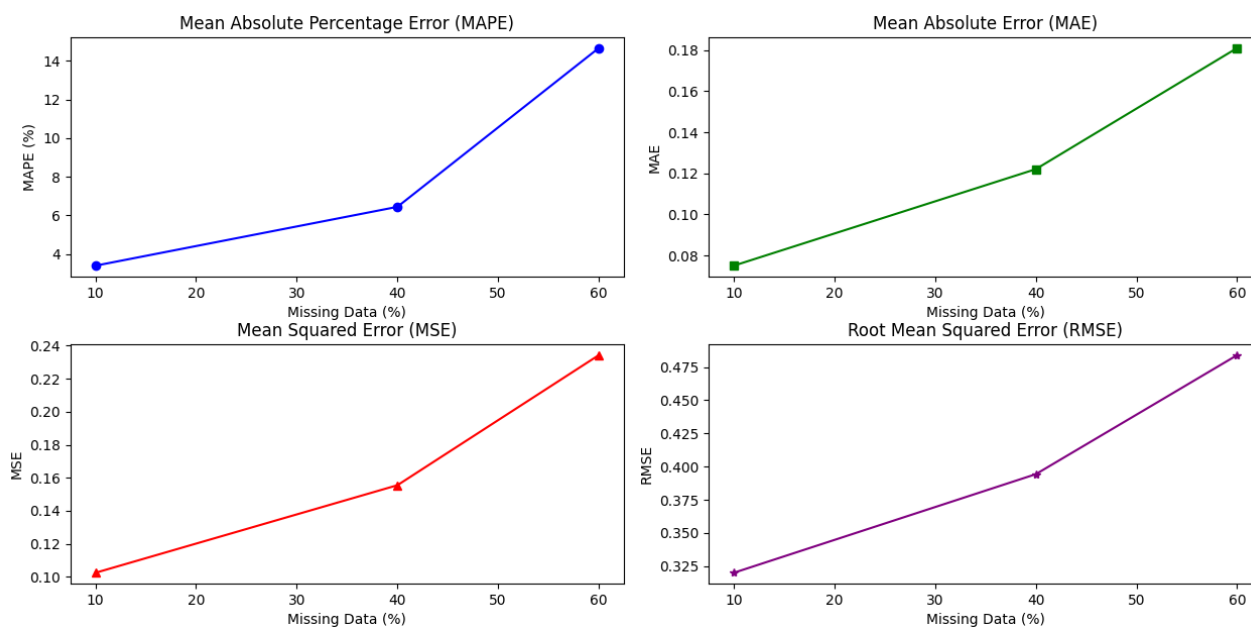


Figure 5.24: Graph of LSTM across different levels of missing Data

The LSTM model attained a loss of 0.079, signaling its adeptness in minimizing the disparity between predicted and observed values. This indicates the model's propensity for closely approximating real-world data (*see figure 5.25*).

Furthermore, these outcomes underscore the model’s proficiency in accurately handling missing values. Overall, the assessment results underscore the effectiveness and promise of the LSTM model in resolving the specified issue, affirming its capacity to provide precise predictions.

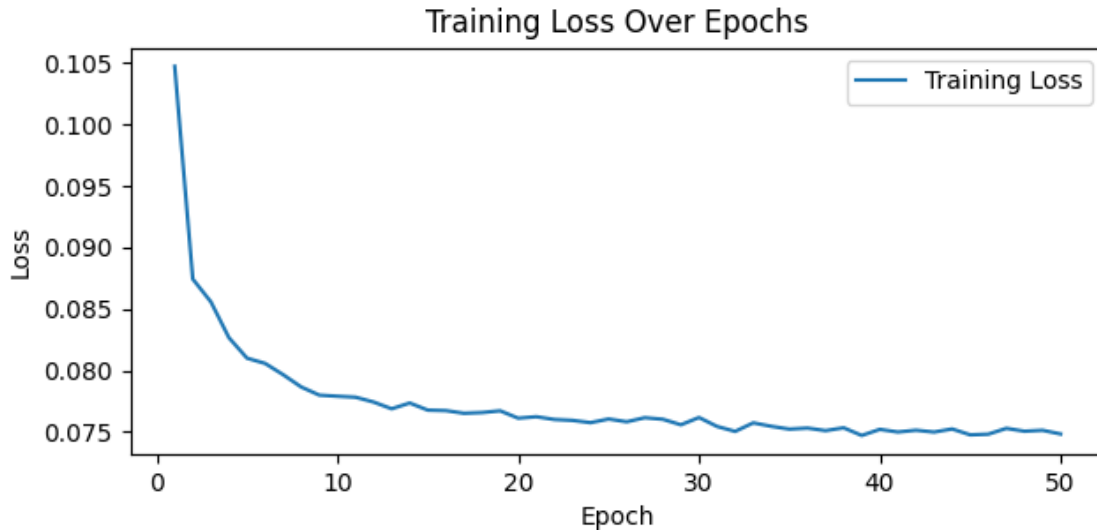


Figure 5.25: Training loss over epochs

### Hybrid (RF\_LSTM) model

In our system, we utilized a hybrid model combining Random Forest (RF) and Long Short-Term Memory (LSTM) networks to handle missing data, leveraging the strengths of both approaches.

This dual approach ensures a more accurate and robust imputation process, improving the overall data quality and model performance. The detailed methodology for implementing this hybrid model is explained in the previous chapter 4, and the results showcasing the efficacy of the RF\_LSTM hybrid model in dealing with missing data are illustrated in table 5.6 and figure 5.26.

Table 5.6: RF\_LSTM across different levels of missing data

Method	Missing Data (%)	MAPE (%)	MAE	MSE	RMSE
RF_LSTM	10	2.675	0.025664	0.0346	0.17609
	40	3.757	0.035904	0.04164	0.20121
	60	5.0536	0.041376	0.04436	0.21316

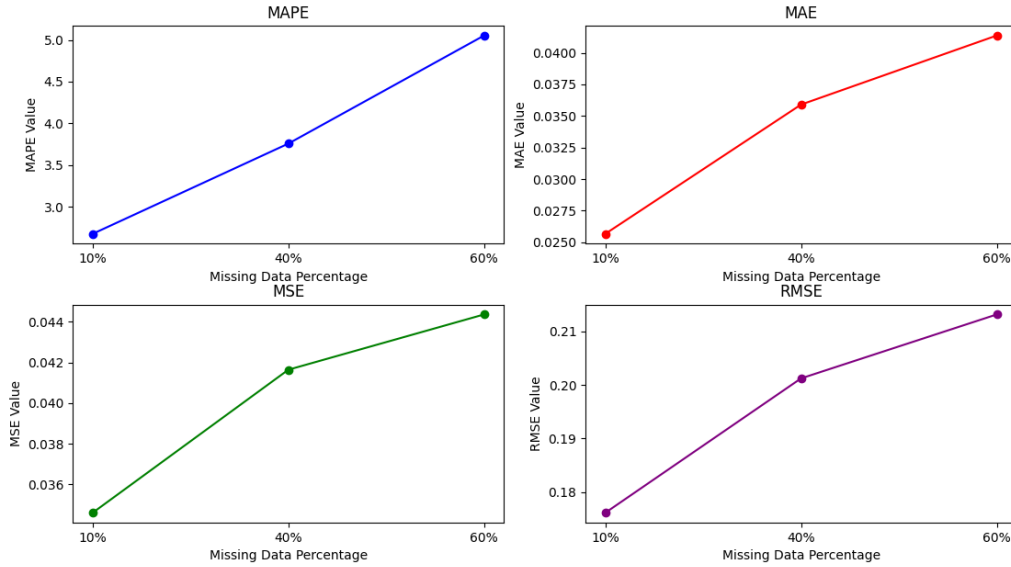


Figure 5.26: Graph of RF\_LSTM across different levels of missing data

The hybrid model RF\_LSTM performed better than RF and LSTM models for different proportions of missing data, where RF had strong performance at low percentages of missing data but this dropped off with an increase in lost data, while LSTM shows a consistent but comparatively higher error rate across all percentages. RF\_LSTM, leveraging the complementary strengths of RF and LSTM, consistently outperforms both RF and LSTM in terms of MAPE, MAE, MSE, and RMSE, demonstrating its effectiveness in mitigating the impact of missing data and improving predictive accuracy (*see figure 5.27*).

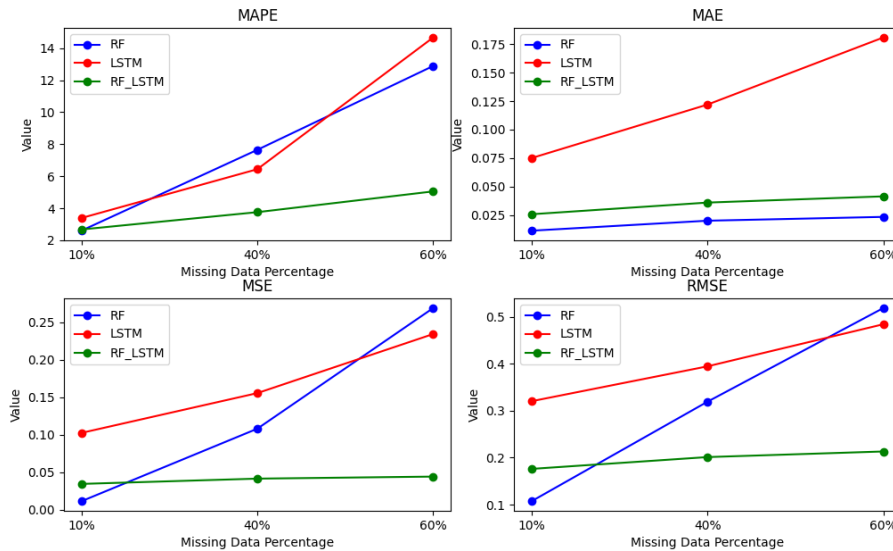


Figure 5.27: Graph comparing other models with RF\_LSTM Across Different Levels of Missing Data

### Advantages of the hybrid model

Hybridizing random forest (RF) and long short-term memory (LSTM) models to handle



missing data compared to using each model independently can offer several advantages:

- **Complementary strengths:** RF and LSTM models have different strengths, where RF is a powerful algorithm for dealing with data with non-linear relationships and missing values, while LSTM is adept at capturing sequential patterns and dependencies in time series data. By combining these two models, we can improve imputation accuracy and benefit from both advantages.
- **Reducing model-specific biases:** Each model may have its own biases and limitations, such that RF may have difficulty capturing long-term dependencies while LSTM may override short-term patterns. By combining these models, we can mitigate model biases and obtain more balanced imputation results.
- **Flexibility:** Hybrid models provide flexibility in model selection and combination that enables us to try different combinations of algorithms, weights, and preprocessing techniques to find the optimal configuration for the dataset.

## 5.7 Discussion

The obtained results clearly show that the hybrid model performs in an effective way the assessment and handling of missing data compared to other models and methods, particularly in scenarios involving various types of missingness and data distributions. The high accuracy and efficiency observed in imputing missing values indicate the effectiveness of our methodology in efficiently estimating the absent data values. This implies that our approach is robust and reliable in addressing missing data challenges, showcasing its potential for practical applications in diverse datasets. To ensure a realistic comparison, we benchmarked our method against the closest works. As we can see in table 5.7, the large difference in metrics indicates a significant difference in the performance of the chosen imputation method across the various datasets. This difference is primarily due to the distinct characteristics of each dataset, such as variability, data distribution, and the nature of the missing data, all of which can affect the performance of the imputation technique. Specifically, it suggests that our dataset has more complexity and higher variability. On the other hand, the positive outcomes consolidate the successful performance of our approach in handling missing data across different scenarios and provide assurance in its capability to enhance data completeness and reliability.

To summarize, our approach has addressed several key challenges in missing data handling and has demonstrated the following strengths:

- Our software tool exhibits the ability to impute missing values accurately and efficiently, ensuring enhanced data quality and completeness.
- The developed [H2MD](#) integrates a diverse array of methods, allowing for comprehensive handling of missing data across various scenarios and datasets.

- The system demonstrates high precision in imputing missing values, minimizing estimation errors, and preserving the integrity of the dataset.

However, like any developed methodology, our software application encountered certain challenges that require further exploration and refinement. These challenges include:

- **The scalability challenge:** The computational resources required for implementing our approach may be substantial, particularly for large datasets or complex missing data patterns.
- **Intrinsic properties of missing data:** Our software application may encounter difficulties in accurately imputing missing values in cases where missingness is dependent on unobserved factors or intricate data relationships.
- **Data sparsity issues:** In some instances, our software application may struggle to handle missing data in categorical variables with high cardinality or in datasets with sparse observations, potentially leading to suboptimal imputation outcomes.

Table 5.7: Comparison of research study techniques using different approaches

Authors	Year	Dataset	Type of missing data	Missing data percentage	Method	Metrics			
						RMSE	MAPE (%)	MAE	MSE
Our approaches	2024	weather	undefined	10	Mean	308.00	56.1	831.06	94864.96
					Median	292.56	62.8	710.16	85588.84
					Mode	261.71	50.2	650.46	68493.25
					Random forest	2.60	0.1072	0.0112	0.0115
					LSTM	0.3201	3.39	0.075	0.1025
					RF_LSTM	0.17609	2.675	0.025664	0.0346
[80]	2023	water station	MCAR	40	Seasonal decomposition	5.25	4.32	undefined	undefined
[81]	2020	Forest fires	MCAR	45	TSIF-AL	83	36.33	undefined	undefined
					KNNI	92.24	66.19	undefined	undefined
[82]	2018	Airline passenger	undefined	undefined	Mean imputation	96.91	29.22	83.43	undefined
					Regression Imputation	22.34	9.18	17.92	undefined

## 5.8 Conclusion

This chapter outlines a comprehensive approach to handling missing data, incorporating multiple techniques. Our software application demonstrates accuracy and efficiency in imputing missing values across diverse datasets. Precision minimizes estimation errors and ensures the integrity of the dataset. By leveraging advanced techniques such as random forest, LSTM, and other techniques, we enhance imputation accuracy and produce robust datasets for subsequent analyses. Further, we have emphasized the importance of adopting a multifaceted approach to effectively address missing data challenges.

Despite the existing challenges, our approach offers a promising framework for addressing missing data effectively and enhancing data quality in practical applications. Future research efforts could focus on optimizing computational efficiency, enhancing robustness to complex missing data patterns, and exploring strategies for handling missing data in specific data types or domains.

# General conclusion

Dealing with missing data represents a critical challenge for every data analyst and information system manager. Through this work, we explored a set of techniques that aim to effectively handle missing data that combine imputation and deletion methods and artificial intelligence models. We first investigated statistical techniques such as mean, mode and median. We found that this type of technique provides simplicity and computational efficiency, but may oversimplify the distribution of the underlying data and lead to biased estimates, especially in the presence of complex patterns or significant missingness. Followed by works on deletion techniques that include listwise deletion and pairwise deletion, entire variable deletion. Although these methods are easy to implement, they may result in a significant loss of valuable information, which may lead to low model accuracy and biased results.

To overcome the limitations imposed on traditional methods, we introduced machine learning models into our framework. Random Forest has proven its effectiveness in imputing missing values by focusing on the relationships between variables in the dataset. A set of decision trees is generated, so Random Forest can capture complex interactions, and consequently, impute missing values more accurately than simpler methods.

In addition, we incorporated long short-term memory (LSTM) networks (*a type of recurrent neural network (RNN)*) into our hybrid model as they excel at capturing temporal dependencies, making them particularly useful for missing sequential data.

Our developed software tool has proven its superior performance in attributing missing data through extensive testing, demonstrating high levels of accuracy and efficiency that make it suitable for various real-world applications.

Furthermore, the developed H2MD tool shows its superior performance in imputing missing data through extensive testing, demonstrating high levels of accuracy and efficiency that make it suitable for various real-world applications. As a direction for our future work, we have identified several perspectives and goals to further enhance our system for handling missing data by:

- Training the model using a large dataset originating from various domains.
- Train the model using a dataset containing different data formats such as images and natural language data.

By focusing on these perspectives in our future work, we aim to increase the performance and capabilities of our system, by focusing on the following research directions.:

- Explore techniques for imputation of missing data in multilingual corpora to ensure consistency across different languages.
- Design adaptive imputation methods that dynamically choose the best imputation strategy based on data characteristics.
- Incorporate user feedback and domain expertise into the imputation process, enabling interactive optimization of missing data attribution.
- Expanding future work to include different fields such as health, universities, energy areas, and others.
- Expanding future work to include various types of data, such as images, natural language data, and videos.

# Bibliography

- [1] Arkadiusz Krysik. A comprehensive guide to database management systems, June 2023. Accessed on 2024-03-21.
- [2] Undisclosed Writer. Continuation of introduction to network model, December 2023. Accessed on 2024-03-27.
- [3] Sakhi Bhagwat. What is object-oriented model in dbms?, May 4 2023. Accessed on 2024-03-27.
- [4] Matthew Tyson. What is mongodb? a quick guide for developers, July 2021. Accessed on 2024-03-11.
- [5] DevOps. Apache cassandra overview, October 2016. Accessed on 2024-03-21.
- [6] Siddhartha Sehgal. From understanding to setup — installing neo4j on an azure virtual machine (linux/ubuntu), December 2019. Accessed on 2024-03-21.
- [7] Merissa Badenhorst. An introduction to centralised databases, July 2021. Accessed on 2024-03-11.
- [8] What is a distributed database?, May 2021. Accessed on 2024-03-11.
- [9] Cloud database vs. traditional database. Accessed on 2024-03-11.
- [10] Utkarsh. Handling missing values - categorical & numerical, May 2023. Accessed on 2024-03-21.
- [11] Binay Gupta. Dealing with unclean data - imputing missing values, May 2023. Accessed on 2024-03-21.
- [12] Kunal Makwana. Frequent category imputation (missing data imputation technique), May 2021. Accessed on 2024-03-21.
- [13] Jacob Joseph. How to treat missing values in your data : Part ii, April 2016. Accessed on 2024-03-21.

## BIBLIOGRAPHY

---

- [14] S. Wang, B. Li, M. Yang, and Z. Yan. Missing data imputation for machine learning. In *IoT as a Service: 4th EAI International Conference, IoTaaS 2018, Xi'an, China, November 17–18, 2018, Proceedings*, pages 67–72. Springer International Publishing, 2019.
- [15] Pedro J. García-Laencina, José Luis Sancho-Gómez, and Aníbal R. Figueiras-Vidal. Pattern classification with missing data: a review. *Neural Computing and Applications*, 19:263–282, 2010.
- [16] Diagrams for Businesses Diagrams for Software Engineering. What is decision tree analysis? how to create a decision tree, March 2021. Accessed on 2024-03-21.
- [17] Andre Ye. Missforest: The best missing data imputation algorithm?, August 2020. Accessed on 2024-05-04.
- [18] LogiqueTechno. 7 conseils aux débutants pour apprendre python. Accessed on 2024-04-08.
- [19] Marina Wyss. Understanding and handling missing data, March 31 2020. Accessed on 2024-03-20.
- [20] Amazon Web Services. Qu'est-ce que la science des données?, Accessed on 2024-02-22.
- [21] reintechno.io. Data analysis, Unknown.
- [22] Zaur Rasulov. Missing values in data science, February 21 2021.
- [23] MongoDB. Types of databases, Accessed on 2024-02-19.
- [24] javatpoint. Types of databases, Accessed on 2024-02-13.
- [25] Sergey V. Kholod. Performance comparison for different types of databases. 2021.
- [26] Omar Alotaibi and Eric Pardede. Transformation of schema from relational database (rdb) to nosql databases. *Data*, 4(4):148, 2019.
- [27] Neha Thakur and Nidhi Gupta. Relational and non-relational databases: A review. *Journal of University of Shanghai for Science and Technology*, 23(8):117–121, 2021.
- [28] phoenixNAP. What is an object-oriented database, April 15 2021.
- [29] Saptarshi Mukherjee. The battle between nosql databases and rdbms. 2019. Available at SSRN 3393986.
- [30] Bauer College of Business, University of Houston. Missing data: the hidden problem draw more valid conclusions with spss missing data analysis. White Paper. <https://www.bauer.uh.edu/jhess/documents/2.pdf>.
- [31] Zhi Li. Nosql databases. *The Geographic Information Science & Technology Body of Knowledge (2nd Quarter 2018 Edition)*, 2018.
- [32] Adrienne Watt. Chapter 6 classification of database management systems. Accessed on 2024-03-11.

- [33] Dmitry Ermakov. Cloud-based vs traditional databases: which one should you choose?, May 2023. Accessed on 2024-03-11.
- [34] Pritha Bhandari. Missing data | types, explanation, & imputation, December 8 2021. Revised on June 21, 2023.
- [35] Shahid Alam, Muhammad S. Ayub, Sanchit Arora, and Muhammad A. Khan. An investigation of the imputation techniques for missing values in ordinal data enhancing clustering and classification analysis validity. *Decision Analytics Journal*, 9:100341, 2023.
- [36] I-COM Global. *Emerging Issues in Data Storytelling*, 2021. Comments Copyright © I-COM Global 2021. Structural: The Challenges of Working with Incomplete Data Sets. Source: Kantar, Getting Media Right, 2019.
- [37] Jos W. R. Twisk. *Applied Longitudinal Data Analysis for Epidemiology*. Cambridge University Press, 5 2013.
- [38] Ahmed M. Gad and Abdel-Salam S. Ahmed. Analysis of longitudinal data with intermittent missing values using the stochastic em algorithm. *Computational Statistics & Data Analysis*, 50(10):2702–2714, 2006.
- [39] YData. Understanding missing data mechanisms: Types and implications, 7 2023.
- [40] Jon Arni Steingrímsson, David H. Barker, Robert Bie, and Ioannis J. Dahabreh. Systematically missing data in causally interpretable meta-analysis, 2022. Accessed on 2024-03-11.
- [41] Nicole Laskowski and Linda Tucci. What is artificial intelligence (ai)? everything you need to know. CIO/IT Strategy, April 15 2024. Accessed on 2024-04-28.
- [42] What is machine learning (ml)? Accessed on 2024-04-30.
- [43] What is deep learning? Accessed on 2024-04-30.
- [44] Nasima Tamboli. Effective strategies for handling missing values in data analysis, July 14 2023. Accessed on 2024-03-25.
- [45] Guanlan Xu. Pairwise deletion v.s. listwise deletion, July 30 2020. Accessed on 2024-03-20.
- [46] Mike Nguyen. *A Guide on Data Analysis*. 2020. Accessed on 2024-03-20.
- [47] Arthur C. Codex. Missing data imputation with r, September 13 2023. Accessed on.
- [48] Sik-Yum Lee. *Handbook of Latent Variable and Related Models*. Elsevier, 2011.
- [49] D. Shumeiko and I. Rozora. Handling missing values in machine learning regression problems. In *IntSol Workshops*, pages 211–219, 2021.
- [50] S. M. Mostafa. Imputing missing values using cumulative linear regression. *CAAI Transactions on Intelligence Technology*, 4(3):182–200, 2019.
- [51] T. Emmanuel, T. Maupong, D. Mpoeleng, T. Semong, B. Mphago, and O. Tabona. A survey on missing data in machine learning. *Journal of Big Data*, 8(1):1–37, 2021.



- [52] P. Dufossé and S. Benzekry. Une comparaison des algorithmes d'apprentissage pour la survie avec données manquantes. 2023.
- [53] S. Wang, B. Li, M. Yang, and Z. Yan. Missing data imputation for machine learning. In *IoT as a Service: 4th EAI International Conference, IoTaaS 2018, Xi'an, China, November 17–18, 2018, Proceedings*, volume 4, pages 67–72. Springer International Publishing, 2019.
- [54] Pedro J García-Laencina, José L Sancho-Gómez, and Aníbal R Figueiras-Vidal. Pattern classification with missing data: a review. *Neural Computing and Applications*, 19:263–282, 2010.
- [55] Felix A Gers, Jürgen Schmidhuber, and Fred Cummins. Learning to forget: Continual prediction with lstm. *Neural computation*, 12(10):2451–2471, 2000.
- [56] T. Emmanuel, T. Maupong, D. Mpoeleng, T. Semong, B. Mphago, and O. Tabona. A survey on missing data in machine learning. *Journal of Big Data*, 8(1):1–37, 2021.
- [57] H. Ou, Y. Yao, and Y. He. Missing data imputation method combining random forest and generative adversarial imputation network. *Sensors*, 24(4):1112, 2024.
- [58] Sunil Kumar Dash. Handling missing values with random forest, 2022. Accessed on 2022-09-22.
- [59] S. Jäger, A. Allhorn, and F. Bießmann. A benchmark for data imputation methods. *Frontiers in Big Data*, 4:693674, 2021.
- [60] H. Jiang, C. Wan, K. Yang, Y. Ding, and S. Xue. Continuous missing data imputation with incomplete dataset by generative adversarial networks–based unsupervised learning for long-term bridge health monitoring. *Structural Health Monitoring*, 21(3):1093–1109, 2022.
- [61] T. F. Johnson, N. J. Isaac, A. Paviolo, and M. González-Suárez. Handling missing values in trait data. *Global Ecology and Biogeography*, 30(1):51–62, 2021.
- [62] S. Bähr, G. C. Haas, F. Keusch, F. Kreuter, and M. Trappmann. Missing data and other measurement quality issues in mobile geolocation sensor data. *Social Science Computer Review*, 40(1):212–235, 2022.
- [63] I. Izonin, R. Tkachenko, V. Verhun, and K. Zub. An approach towards missing data management using improved grnn-sgtm ensemble method. *Engineering Science and Technology, an International Journal*, 24(3):749–759, 2021.
- [64] Y. Zhang and P. J. Thorburn. Handling missing data in near real-time environmental monitoring: A system and a review of selected methods. *Future Generation Computer Systems*, 128:63–72, 2022.
- [65] D. X. Yang, R. Khera, J. A. Miccio, V. Jairam, E. Chang, B. Y. James, and S. Aneja. Prevalence of missing data in the national cancer database and association with overall survival. *JAMA network open*, 4(3):e211793–e211793, 2021.

## BIBLIOGRAPHY

---

- [66] Z. Xue and H. Wang. Effective density-based clustering algorithms for incomplete data. *Big Data Mining and Analytics*, 4(3):183–194, 2021.
- [67] M. Mutasim and A. Karrar. Impute missing values in r language using ibk classification algorithm. *International Journal of Engineering Science and Computing*, 11(6):28328–28338, 2021.
- [68] Y. Kim, S. Steen, and H. Muri. A novel method for estimating missing values in ship principal data. *Ocean Engineering*, 251:110979, 2022.
- [69] Intellipaat. What is pycharm?, December 2023. Accessed on 2024-04-08.
- [70] GeeksforGeeks. History of python, December 2023. Accessed on 2024-04-08.
- [71] GeeksforGeeks. What is pytorch?, 2023. Accessed: 2024-05-28.
- [72] Tutorialspoint. Scikit learn - introduction. Accessed on 2024-04-12.
- [73] CodersLegacy. PyQt6 Tutorial Series | Python GUI Programming. Accessed on 2024-04-12.
- [74] GeeksforGeeks. Pandas introduction, March 2024. Accessed on 2024-04-12.
- [75] Matplotlib Development Team. Matplotlib: Visualization with python. Accessed on 2024-04-12.
- [76] NumPy Developers. Numpy: the absolute basics for beginners. Accessed on 2024-04-12.
- [77] Ravikiran A S. Python gui: Build your first application using tkinter, June 2023. Accessed on 2024-04-12.
- [78] GeeksforGeeks. Mysql-connector-python module in python, March 09 2020. Accessed on 2024-05-03.
- [79] Davide Chicco, Matthijs J Warrens, and Giuseppe Jurman. The coefficient of determination r-squared is more informative than smape, mae, mape, mse and rmse in regression analysis evaluation. *PeerJ Computer Science*, 7:e623, 2021.
- [80] X. Lai, L. Zhang, and X. Liu. Takagi-sugeno modeling of incomplete data for missing value imputation with the use of alternate learning. *IEEE Access*, 8:83633–83644, 2020.
- [81] N. Umar and A. Gray. Comparing single and multiple imputation approaches for missing values in univariate and multivariate water level data. *Water*, 15(8):1519, 2023.
- [82] I S Iwueze, E C Nwogu, V U Nlebedim, U I Nwosu, and U E Chinyem. Comparison of methods of estimating missing values in time series. *Open Journal of Statistics*, 8(2):390–399, 2018.

# Annex

## Startup creation

### 1.1 Project presentation

#### 1.1.1 The project idea (the proposed solution)

Handling missing data that arises for various reasons in datasets is crucial to ensuring the accuracy and reliability of data analyses and for better informed decision-making. Indeed, the data sets (databases) managed by government organizations (hospitals, universities, administrations, ministries, etc.) or those of the economic sector (agricultural sector, energy, commerce, etc.) cannot escape the existence of certain values that are expected to be collected but that are empty or missing. This phenomenon is due to several reasons, such as: human factors (entry errors), technical problems (sensor or software failures), absence of responses in the questionnaires (surveys, population resurgence, etc), natural causes (external events, historical data, data protected by law, etc).

The initiative focuses on overcoming challenges inherent in incomplete datasets due to manual data collection and storage inefficiencies. By applying statistical and deletion AI techniques, our goal is to tackle the problem by estimating and assessing the missing values. Hence, we effectively fill in the missing data while preserving the statistical integrity of the considered dataset. Additionally, leveraging AI techniques, particularly machine learning models capable of handling missing data gracefully, offers advanced solutions. These approaches not only enhance data completeness but also bolster the robustness and reliability of analytical outcomes within economic and government firms, characterized by a massive absence of reliable data at the source, such as data on employment, agriculture, the parallel economy, and the informal sector. The implementation of the proposed solution in a software tool involves rigorous testing and validation phases tailored to local data characteristics, ensuring optimized performance and compatibility across various real-world datasets. Through iterative refinement based on local feedback and requirements, our approach aims to enrich and improve data quality, strengthen decision-making processes, and facilitate advancements across various sectors within Algeria's state organizations, economic and financial firms, as well as private companies.

### 1.1.2 The suggested values

- **Innovation:** Our project [H2MD](#) integrates advanced statistical, deletion, and AI techniques to address missing data challenges, revolutionizing data preprocessing methodologies.
- **Performance:** We ensure robust performance by employing efficient algorithms that enhance data integrity and analytical outcomes.
- **Cost efficiency:** Automated data handling reduces operational costs associated with manual data cleaning and preparation.
- **Flexible solutions:** Designed to fit diverse data structures and analysis requirements, our interface provides customizable solutions that adapt to specific data challenges and format diversity.
- **Scalability:** Built to scale with growing data volumes and evolving analytical needs, our interface supports seamless integration into existing data ecosystems such as easyPHP.
- **Design:** Intuitive and user-centric, our interface offers an animated and easy-to-navigate visualization, tailored to the needs of analysts and data scientists.

### 1.1.3 The working team

**Bouressace Kawkab:** A computer science student who has already developed several websites and applications. Her skills include Python, Java, and C, JavaScript, HTML, and CSS. She is responsible for the design, development, and maintenance of [H2MD](#)'s software application, ensuring the website functions properly.

**Dr. Khebizi Ali:** A senior lecturer at the Computer Science Department of 8 May 1945 Guelma University and a permanent researcher at the LabSTIC laboratory. With extensive experience in databases, information systems, data analysis, machine learning, and artificial intelligence, Dr. KHEBIZI provides invaluable guidance and supervision for our project. His rich expertise in various applicative domains (Wilaya de Guelma, CNR, social security agencies, and private firms) ensures that the conceived framework and the developed software tool are scientifically sound, and their feasibility and applicability can contribute in an efficient manner to resolving real-world data challenges. Dr. KHEBIZI's mentorship helps drive the project forward, ensuring that we achieve our objectives with the highest standards of academic and practical rigor.

### 1.1.4 Project development and deployment plan

#### Short-term (1 year):

**Pilot Implementation:** Target a few representative sectors to test the project deployment with real data. Launch a pilot project in certain sectors or departments to demonstrate effectiveness and obtain initial feedback from users.

**Achieve Adoption:** Aim for a 15% adoption rate among targeted users to validate effectiveness and refine implementation strategies.

**Medium-term (2-3 years):**

**Expansion and Integration:** Scale implementation across diverse sectors or departments, expanding to cover a wider range of data types and challenges.

**Enhance Functionality:** Incorporate feedback and iterate on the system kernel to integrate additional features and functionalities based on user requirements and emerging trends.

**Long-term (4-5 years):**

**Market Leadership:** Establish market leadership by capturing a significant share of the target market, becoming the preferred solution for comprehensive data handling needs. **Global Expansion:** Evaluate opportunities for international expansion, particularly in the Maghreb and Africa with similar geographical data (climate, forests, agriculture, etc.), where completed data can be generalized and regulatory environments can be unified (Maghreb).

**1.1.5 The project timeline**

Our team meticulously planned the different phases of the development of the software application (H2MD). The following timeline details the main milestones of the project, highlighting our commitment to delivering a robust solution within the set deadlines.

Table 1.1: Project tasks timeline: (Mo) Month

No	Task description	Mo 1	Mo 2	Mo 3	Mo 4	Mo 5	Mo 6	Mo 7
1	Research on missing data handling techniques	✓	✓					
2	Development of statistical imputation methods		✓	✓				
3	Implementation of deletion techniques		✓	✓				
4	Integration of AI algorithms			✓	✓	✓		
5	Design and development of animated interface				✓	✓	✓	
6	Comparative analysis of methods					✓	✓	
7	Final testing and validation							✓

This timeline shows in table 1.1 the key stages of the development of H2MD. Each phase has been carefully planned to ensure the project is carried out in the best possible conditions and on time, thus maximizing the chances of success from the outset.

## 1.2 Innovative aspects

### 1.2.1 The nature of innovations

#### Market innovations

"H2MD" addresses a critical need in data analysis in Algeria by offering an integrated software application for effective management of incomplete data. This introduces new opportunities in a growing Algerian streamlined governance and a knowledge-based economy, where data quality and informed decision-making are the key concepts for quick economic emergence on a global scale.

#### Technological innovations

The "H2MD" software application integrates advanced techniques for handling missing data, including robust statistical and deletion methods, artificial intelligence techniques. The animated interface provides dynamic visualization of data processing workflows, enhancing understanding and enabling quick decision-making for users.

#### Growth innovations

"H2MD" is designed to evolve with the changing needs of users in Algeria. By integrating AI techniques for imputing missing data and comparing results with other methods such as data deletion, the platform offers personalized recommendations based on comprehensive analysis. This helps optimize strategic and operational decisions effectively.

#### Handling uncertainties

- **Market Uncertainty:** Comprehensive market studies and an initial pilot program will assess the responsiveness of the Algerian market to "H2MD".
- **Technological Uncertainty:** Continuous adoption of new technologies ensures the reliability, and performance of the software application.

## 1.3 Areas of innovation

### 1.3.1 New Processes

"H2MD" enhances operational efficiency and data reliability by integrating advanced statistical and AI techniques, optimizing data handling processes and reducing uncertainties in decision-making.

### 1.3.2 Enhanced functionalities

The software application introduces dynamic visualization tools and interactive interfaces for exploring various missing data handling techniques, improving user engagement and facilitating

informed decision-making.

### **1.3.3 New clients**

"H2MD" targets new client segments by offering tailored solutions for diverse governmental and economic stakeholders in Algeria, ensuring scalable and effective data management solutions.

### **1.3.4 New offers**

Our hybrid AI model provides innovative solutions for handling missing data, ensuring secure and efficient data management practices that meet the evolving needs of Algerian enterprises and institutions.

### **1.3.5 New models**

"H2MD" revolutionizes traditional data management models by integrating AI-driven analytics and robust statistical and deletion techniques, enabling proactive decision-making and strategic planning in dynamic business environments.

## **1.4 Strategic market analysis**

### **1.4.1 Market segment**

#### **Potential market**

The potential market for "H2MD" spans across various state organizations, economic firms, and private companies in Algeria, as well as the regional environment. These entities require robust data management solutions to improve operational efficiency, decision-making processes. Many of these organizations still rely on outdated or manual data handling methods, presenting a significant opportunity for "H2MD" to introduce modern.

#### **Target market**

Our specific target market includes state universities, hospitals, and municipal administrations needing strategic decision-making and that demonstrate a readiness. These organizations possess the infrastructure and budget to invest in digital technologies that enhance data accuracy, accessibility, and security. They are motivated by the need to streamline operations, comply with regulatory standards, and improve service delivery to stakeholders.

#### **Market selection**

We selected this target market because state organizations typically manage large volumes of sensitive data and face increasing pressure to adopt efficient and political and strategic decision-making. They often seek solutions that not only address immediate operational challenges but also

support long-term scalability and compliance with national and international economic evolution and trends.

## 1.4.2 Measuring competition intensity

### Direct and indirect competitors

"H2MD" faces direct competition from other digital library management solutions operating in the Algerian market. Indirect competitors consist of traditional manual management systems still widely used.

### Strengths and weaknesses of competitors

#### Strengths of competitors:

- Benefit from prior market recognition.
- Established relationships with institutions.

#### Weaknesses of competitors:

- Technology may be outdated.
- User interface may not meet modern expectations.
- Lack of innovative features.
- Reliance on older, less adaptable solutions.
- Slower adaptation to current needs.
- Higher costs associated with upgrades and maintenance.

## 1.4.3 Marketing strategy

- **Promotional offers:** Regularly offering promotions and loyalty programs to incentivize adoption of advanced data management solutions. These initiatives not only attract new clients but also foster long-term relationships by demonstrating the value of improved data integrity and analysis.
- **Educational workshops:** Hosting workshops and webinars to educate potential clients in Algeria about the importance of effective data management practices and the benefits of adopting "H2MD" solutions by exploiting its functionalities.
- **Partnership development:** Collaborating with local universities, hospitals, municipalities, and other state organizations to pilot and implement "H2MD" Solutions. These partnerships not only validate the effectiveness of the platform but also serve as references and testimonials for attracting new clients.



- **Continuous innovation:** Committing to continuous research and development to enhance "H2MD" Solutions. This ongoing innovation ensures that the platform remains competitive and capable of meeting evolving data management needs in the Algerian economic ecosystem..

## 1.5 Financial plan

To effectively manage and process datasets with missing data, it's essential to outline all costs and investments required accurately. This includes initial setup costs, and ongoing costs. Here are the key elements to consider:

### 1.5.1 Costs and charges

**Detailed cost breakdown:** The costs of the "H2MD" project include software development, hardware acquisition, training, marketing, and administrative expenses. Here's a breakdown of the main budgetary items:

- **Software development:** Costs associated with designing, developing, and implementing the platform.
- **Hardware and infrastructure:** Investments in necessary IT equipment to support the platform.
- **Marketing and advertising:** Budget for promoting and raising awareness about the product.
- **Administrative and overhead costs:** Ongoing costs for managing the business.

**Sources of financing:** To fund these costs, "H2MD" plans to utilize a combination of:

- **Equity investments:** Involvement of investors and financial partners.
- **Grants and aid:** Seeking available grants for technological startups.

### 1.5.2 Revenue forecast

It's crucial to evaluate both optimistic and pessimistic revenue scenarios to forecast the future financial performance of "H2MD". Here is a detailed presentation of sales forecasts for the next five years (*see table 1.2*):

Table 1.2: Subscription data: one year (N)

Year	Number of Subscriptions	Price per Subscription	Total Revenue
N	50	20,000 DZD	1,000,000 DZD
N+1	120	20,000 DZD	2,400,000 DZD
N+2	250	20,000 DZD	5,000,000 DZD
N+3	350	20,000 DZD	7,000,000 DZD
N+4	420	20,000 DZD	8,400,000 DZD

This structured data allows for comprehensive analysis of potential financial outcomes, encompassing both conservative and optimistic revenue scenarios.