Downstream and Privacy Preserving Missing Data Recovery for IoT Systems

Benjamin Agbo

School of Computing and Engineering

University of Huddersfield

This dissertation is submitted for the degree of

Doctor of Philosophy

Supervisors: Dr. Hussain Al-Aqrabi

and Prof. Richard Hill

February 2023

To my mother and father,

my wife and,

my siblings,

who made all of this possible,

for their endless encouragement and patience.

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements.

Benjamin Agbo

February 2023

Acknowledgements

I would like to express my heart felt gratitude to key individuals who played a major role in my research journey and the development of this thesis. First and foremost, I would like to appreciate **Dr. Hussain Al-Aqrabi** and **Prof. Richard Hill** for their heartwarming support, advice, guidance and encouragements that enabled me to strive towards completing this thesis. The development of this thesis would not have been possible without their support.

I would also like to thank my dear colleague **Dr. Tariq Alsboui** for his profound encouragements and advice through out the period of my research.

I would like to express my gratitude to my beloved parents **Prof. and Mrs Agbo**, my siblings and family for their support, prayers and encouragement. To all my friends that advised me and offered assistance in diverse ways, I am very grateful.

Last but not least, I would like to express my heart felt gratitude to my wife Deborah for her love and support through out this journey, I'll forever be grateful.

Copyright Statement

List of Publications

1. Agbo, Benjamin, Yongrui Qin, and Richard Hill. "Research Directions on Big IoT Data Processing using Distributed Ledger Technology: A Position Paper." IoTBDS (2019): 385-391.

   Contributions: *In this paper, I developed a framework for secure and private data processing using Distributed Ledger Technology (DLT). I conducted an extensive literature review exploring the merits of an emerging technology called IOTA and highlighted the viability of the IOTA Tangle in supporting data pre-processing tasks.*

2. Agbo, Benjamin, Yongrui Qin, and Richard Hill. "Best Fit Missing Value Imputation (BFMVI) Algorithm for Incomplete Data in the Internet of Things." IoTBDS. 2020.

   Contributions: *In this paper I developed a robust imputation algorithm that is capable of choosing the most plausible technique for imputation from a range of pre-defined sub-techniques. I ran simulations and analysed results to measure the performance of my proposed technique against state-of-the-art missing data imputation techniques. Overall, the proposed technique showed good performance in terms of imputation accuracy and computational complexity.*

3. Agbo, Benjamin, Hussain Al-Aqrabi, Richard Hill, and Tariq Alsboui. "Missing Data Imputation in the Internet of Things Sensor Networks." Future Internet 14, no. 5 (2022): 143.

Contributions: *This paper showcases further improvements made to my proposed missing data imputation technique and evaluated its efficiency in the calibration of on-field low cost sensors for air quality monitoring stations. I applied a suitable methodology and simulated various missing data rates to assess the quality of impuation techniques. Overall, the proposed BFMVI algorithm showed the best performance in terms of imputation accuracy and viability for sensor calibration.*

Abstract

Noticeable growth in the use of intelligent devices has resulted in the generation of vast amounts of data from sensor devices. When dealing with large amounts of data, it is common to observe databases with large amounts of missing values. This is a challenge for data miners because various methods for data analysis only work well on complete databases. A traditional approach to handling missing data is to discard instances of missing values and only use complete cases for analysis. However, research has shown that this approach is not practical especially when large amounts of data are missing. This led to an increased need to develop strategies for replacing missing values with plausible values through imputation. Also, as more sensitive data is also being generated, research has shown the need for more secure and private approaches to pre-processing data. This thesis proposes imputation strategies called $k-BFMVI$ and $med.BFMVI$ for recovering missing values before training downstream regression and classification models respectively. An Average Site Mixture (AvSM) model is further developed to simulate secure missing data recovery for IoT applications using IOTA.

Experiments simulated missingness from 10% to 40% using MCAR and MAR mechanisms. Missing values were further imputed using benchmark techniques and their performance was cross-validated for downstream regression and classification tasks. To simulate distributed settings, missing values were also explored, showing variations in the information held across distributed sites for IoT applications using IOTA.

Overall, the proposed algorithms recorded the best imputation accuracy against benchmark techniques and showed significant improvements on downstream learning.

Table of contents

List of figures

List of tables

Nomenclature

Acronyms / Abbreviations

k-NN k-Nearest Neighbour

$k-BFMVI$ k-Best Fit Missing Value Imputation

CART Classification and Regression Trees

DLT Distributed Ledger Technology

DT Decision Tree

EHR Electronic Health Record

EMI Expectation Maximisation Imputation

IoT Internet of Things

MAE Mean Absolute Error

MAM Masked Authenticated Messaging

MAR Missing at Random

MCAR Missing Completely at Random

ML Machine Learning

ML Regression Imputation

MLR Multiple Linear Regression

NMAR Not Missing at Random

NN Neural Network

PADI Privacy Aware Distributed Intelligence

PI Personal Information

RF Random Forest

RMSE Root Mean Squared Error

TTP Trusted Third Party

UCI University of California Irvine

Chapter 1

Introduction

1.1 Background and Motivation

The Internet of Things (IoT) has played a significant role in the development and adoption of information and communication technologies, penetrating diverse industries and enabling various operations ranging from small scale systems like smart homes and smart healthcare (Javed et al., 2021) to large scale systems such as smart cities, smart manufacturing, autonomous cars, etc., (Suvarna et al., 2020). This has led to an unprecedented increase in the amount of data that is generated and transmitted over the internet. Today, IoT devices ranging from wearable devices to smart sensor devices constantly generate data, which require effective and advanced approaches in their analysis (Izonin et al., 2019). The right choice for data analysis tools as well as optimisation and modelling techniques depend on the quality and quantity of the data generated by sensing devices. As the number of IoT sensing devices increases on a daily basis, it has become important to ensure accurate and timely analysis of data when managing smart systems (Genes, 2018).

In the design of smart systems, it is important to take into account the inconsistencies that may occur in sensed data due to various reasons such as anomalies and sensor failure (Fekade et al., 2017). In addition, the transmission of data aggregated from multiple sensor devices

via communication channels are not always successful, leading to data losses. All these challenges lead to inconsistencies which significantly diminishes the accuracy of insights obtained from IoT data (Izonin et al., 2019).

For instance, some research showed the impact of inconsistent data on large scale power systems such as the U.S. and Canadian electrical failure in 2003 (Burpee et al., 2006) and the 2012 Indian electrical failure (Lai et al., 2013). Therefore, ensuring the quality of IoT data from sensing infrastructures is paramount for making accurate predictions.

According to Genes (2018), limitations in the quality of IoT data are mostly influenced by external factors such as missing data. Research carried out by Genes et al. (2016) showed that lost measurements during data acquisition due to sensor failures, issues with data storage and broken communication are common causes of missing data. The issue of missing data is a common problem that affects many real-world datasets such as traffic data, medical records, industrial applications etc., thereby making it difficult to implement modelling, data analysis and optimization techniques as it introduces uncertainties and bias in a dataset (Okafor, 2021). In light of this, it is important to build missing data estimation techniques that are robust enough to preserve the predictive capabilities of a model.

Numerous techniques have been proposed from various scientific backgrounds to curb this problem. Some of the earliest and widely adopted solutions to this problem are down-sampling, commonly known as complete case analysis, and imputation (Okafor, 2021). Complete case is a simple method that involves discarding observations with incomplete/missing data. However, this method is only suitable for larger datasets and performs better when the missing data rate is significantly low (Osman et al., (2018), Graham, (2009)). On the other hand, appropriate techniques can be applied on a dataset to replace missing instances with plausible values. This method is called imputation.

In addition to the problems caused by missing data, Karkouch et al., (2016) reported other factors that affect the quality of IoT data. Some of which are: Resource constraints, Issues with privacy preservation and Security vulnerability

All these challenges can be identified at different phases of the IoT data cycle. For instance, sensor failures occur at the data acquisition/generation phase while issues with privacy and security occur during data storage, data use or data sharing (Byabazaire et al., (2020); Agbo et al., (2020)).

In view of this, a robust imputation technique for addressing the issue of missing data is proposed in this thesis for downstream learning. This thesis further presents a privacy preserving approach for recovering missing data for distributed systems by exploiting the potential of the IOTA Tangle.

1.2 Research Objectives

This research aims to develop a robust missing data imputation technique for handling downstream machine learning tasks. As security and privacy concerns have grown in the generation and transmission phase, this research further aims to develop a privacy preserving approach to handling missing data. The objectives of this research can be summarised as follows:

1. Examine state-of-the-art missing data approaches and their effects on various rates and mechanisms of missing data.

2. Develop a robust technique for missing data imputation that is capable of choosing from optimal sub-techniques to improve the accuracy of predictions.

3. Evaluate the effect of existing and new imputation algorithms on downstream regression and classification tasks.

4. Develop a secure and private framework for recovering missing data without the need for sharing subject level information in a distributed space.

1.3 Contributions

The contributions of this thesis are summarised as follows:

1. As various imputation techniques exist in literature, when faced with real-life missing instances where there is no ground truth data, it is important to have imputation approaches that will embed the capability of selecting optimal algorithms for imputing missing values. This thesis proposes an imputation algorithm called $k-BFMVI$ that is capable of choosing appropriate techniques for filling-in missing instances based on established features and labels in a given distribution.

2. A detailed demonstration of imputation techniques on synthetic data experiments is presented in this thesis. These experiments showed an in depth statistical view of traditional imputation techniques.

3. A cross-validated method is also developed using a reverse error score function RES(r) that is based on two error calculations between two final imputation estimates, which is used to obtain final imputation results for filling in missing instances.

4. This thesis demonstrates that efficient imputation using $k-BFMVI$ improves the performance of downstream regression tasks.

5. This thesis further develops the $med.BFMVI$, which is an extension of the $k-BFMVI$ and presents a single view of each imputation sub-technique in this approach. The performance of these sub-techniques are further weighed against benchmark techniques. For each sub-technique in $med.BFMVI$, the author demonstrates that selecting the best

plausible sub-technique improves the accuracy of downstream classification tasks in clinical data.

6. To facilitate a more secure and private process for data pre-processing and missing data imputation, This thesis introduces a privacy aware distributed intelligence approach (PADI) using IOTA's Masked Authenticated Messaging (MAM) protocol and demonstrates its viability for ensuring data privacy.

7. An Average Site Mixture (AvSM) algorithm is proposed which leverages the parameter estimates from different sites before imputation without the need for sharing subject level information across data sites.

1.4 Methods

In this thesis, the author follows the recommendations of state-of-the-art research for simulating missing data, paying attention to the principles stipulated by Little and Rubin (1987). To further evaluate the implications of missing data where privacy and security is concerned, the author investigates vertically partitioned data and proposes a privacy-preserving approach for recovering missing data by considering IoT applications using the IOTA Tangle. The following subsections highlight the methods applied in the technical chapters of thesis.

1.4.1 Data Collection and Analysis

The data used for conducting experiments were reviewed and collected from secondary data sources through the University of California Irvine (UCI) machine learning repository. The datasets were carefully validated from the works of previous authors and selected to describe the problem statements considered in this thesis as described in figure 1.1. For the purpose of describing missing data mechanisms and presenting a general view of traditional missing data

imputation techniques, the author also uses a small synthetic dataset with 150 observations for demonstration purposes (see 2.4).

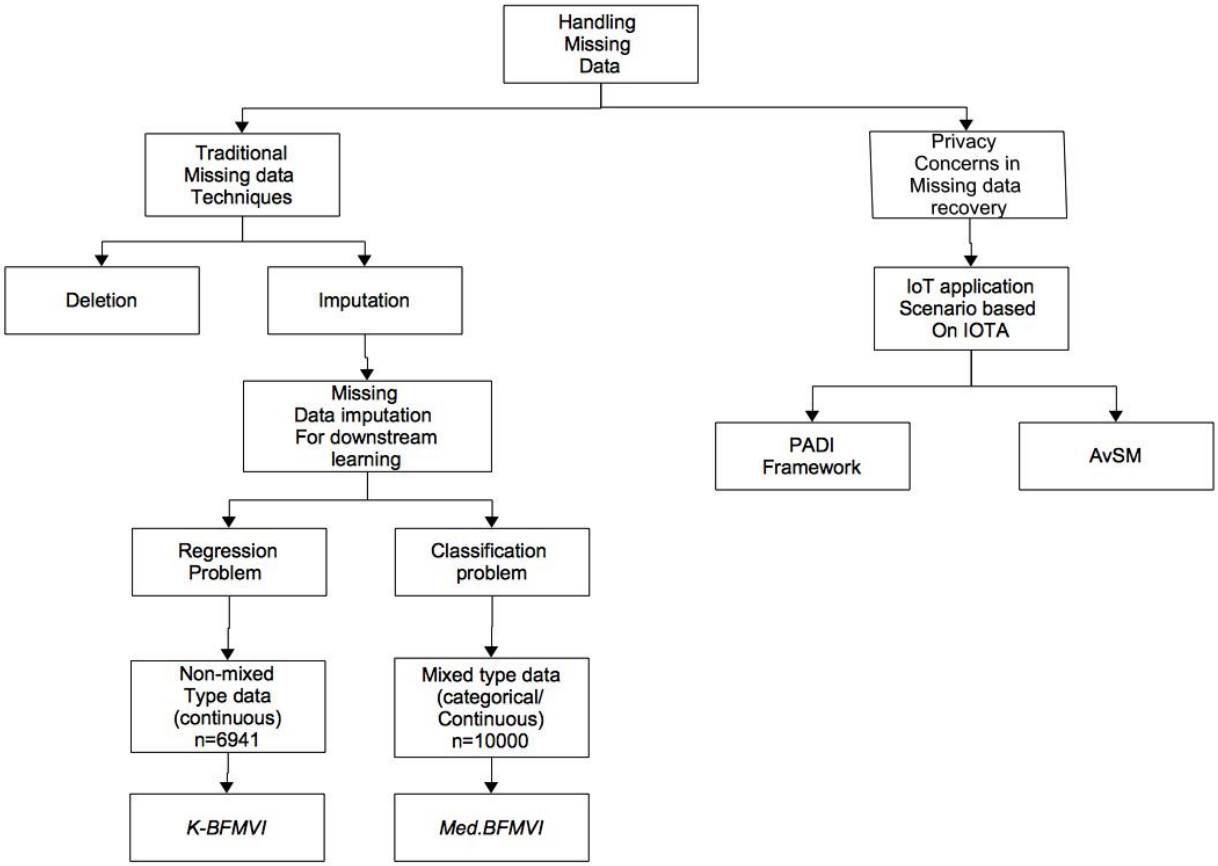

Fig. 1.1 Flow Chart of Research Methods

1.4.2 Simulating Missing Data

In the experiments, the author uses full datasets where all data entries are known. Instances with real missing values were disregarded from analyses because access to the true values of missing instances was required in order to measure the performance of the imputation algorithms. Therefore, artificial missing data was simulated at different rates ranging from 10% to 40% of the overall observations in a distribution. The choice of missing rates was

inspired by the works of Lee et al., (2019) which confirmed that missing data in IoT smart spaces reaches approximately 40% following the analysis of eight streams of IoT data. Missing data was also simulated based on Missing Completely at Random (MCAR) and Missing at Random (MAR) mechanisms.

1.4.3 Performance Evaluation

The performance of each imputation algorithm is evaluated based on the following metrics:

- **Root mean squared error (RMSE) and Mean Absolute Error (MAE)**: This measures the accuracy and precision of the imputation algorithms (how close the predicted values are to the ground truth) and is given by

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{m}(P_{ij} - \hat{P}_{ij})^2} \tag{1.1}$$

and

$$MAE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{m}(|P_{ij} - \hat{P}_{ij}|)} \tag{1.2}$$

where P_{ij} is the original (true) value and $\hat{P}_{ij}$ represents the imputed values. It is important to note that an error score closer to 0 indicates better performance.

- **Classification accuracy**: This is measured by the false negative and false positive error rates (Lee et al., 2002). The false negative error is based on a probabilistic condition that a respondent is classed under a category that is lower than the respondent's true category. Similarly, the false positive error measures the probability of a respondent being classified under a higher category than the respondent's true category. This can be derived by the formula below;

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \times 100 \tag{1.3}$$

- **Coefficient of determination** (R^2): This represents the amount of variance present in a dependent variable that can be predicted from the values of the independent variables, and is given by;

$$R^2 = 1 - \frac{\sum_{i=1}^{m}(P_i - V_i)^2}{\sum_{i=1}^{m}(\bar{p} - P_i)^2} \tag{1.4}$$

  It is noteworthy that a value close to 1 shows better performance.

To further evaluate the performance of imputation techniques, the imputed datasets were further trained using techniques such as; (1) Multiple Linear Regression (2) Decision Trees (3) Random Forest (4) CART and (5) Bootstrap Aggregation. These algorithms are discussed further in the subsequent chapters of this thesis.

1.5 Overview of this Thesis

This thesis proposes a robust technique for missing data recovery that will mitigate the effect of uncertainty in downstream learning tasks for sensor and longitudinal clinical data. For each technique, the author demonstrates their effect when trained on regression and classification algorithms in real-world data sets. In the first part of this research, the author develops and tests approaches which can be applied to sensor networks and wide-spread machine learning problems where missing data occurs. In the second part, the proposed approach is tailored to downstream classification problems in longitudinal clinical studies with uncertain and missing data. In the third part of this thesis, a framework is developed for ensuring the private and secure processing of data using IOTA's MAM protocol. The functionalities of the IOTA tangle is further exploited and a privacy-preserving approach to missing data recovery is further proposed.

A brief outline of the chapters covered in this thesis is presented below:

Chapter 2: Theory of Missing Data and Privacy

In this chapter of the thesis, the author introduces the missing data problem and presents theories and literature around the subject. A review of literature on the security and privacy of IoT data is further presented. Here, a formal discussion is introduced into missing data mechanisms and examples were provided to better understand the concept of these mechanisms. The author further presents the theoretical underpinnings of the missing data mechanisms and discusses the concept of the missing at random mechanism for multi-dimensional data.

Chapter 3: Missing Data Handling

In this chapter, the author identifies traditional and modern missing data handling techniques. A description of the inefficiency of traditional deletion methods is presented, leading to the need for more reliable approaches to handling missing data. This chapter also identifies various imputation techniques as a more suitable solution for handling missing data rather than deletion approaches, which are inefficient when dealing with large amounts of missing data.

Chapter 4: Optimal Missing Data Imputation for Downstream Regression

The $k-BFMVI$ technique is developed in this chapter, which approaches missing data recovery by developing cluster boundaries in the training data and replacing missing values by considering the observed values within each cluster boundary. At this stage, the proposed method is tailored for downstream regression tasks in multivariate data collected from environmental sensor devices. To enhance the selection of the best plausible sub-technique in the $k-BFMVI$ approach, a reverse error score function is proposed to enable the selection of the chosen imputation sub-technique for each partition created in the data set. Different

missing rates are simulated for MAR and MCAR mechanisms and the performance of the proposed technique is analysed against benchmark imputation techniques.

Chapter 5: Imputation of Missing Clinical Covariates for Classification Problems

In this chapter, the author presents the $med.BFMVI$, which is an extension of the $k-BFMVI$ that is designed for recovering missing values in clinical covariates. This technique is tailored for multivariate data with downstream classification problems, which includes data for longitudinal clinical investigations and electronic health records (EHRs). An optimisation formulation that leverages the class dependencies established in the clinical data is considered in the imputation model. Through comparative experiments on real-world clinical data, the author demonstrates that $med.BFMVI$ shows the best performance in terms of imputation accuracy and also shows significant improvements on downstream classification tasks when benchmarked against state-of-the-art techniques.

Chapter 6: Privacy Preserving Approach to Missing Data Imputation for Distributed Systems

In this chapter, a privacy aware distributed intelligent (PADI) approach is developed as a secure and private solution for data pre-processing in distributed systems using IOTA's Masked Authenticated Messaging (MAM) protocol. An Average Site Mixture (AvSM) algorithm is further proposed for recovering missing values across distributed sites using the IOTA Tangle.

Chapter 7: Conclusion and Future Work

This chapter summarises the contributions presented in this thesis and outlines areas of future research.

Chapter 2

Theory of Missing Data and Privacy

In this chapter, a review of missing data theories presented by Little and Rubin (2019) is further highlighted and relevant theories on data privacy in IoT, according to Wahlstrom and Fairweather, (2013) are also identified. The author first of all introduces the theory of missing data by identifying the classes of missing data. A detailed analysis of the concept of missing data is presented based on the missing at random assumption when considering multivariate missing data. A highlight of issues relating to various types of missing data is also presented and their effect on probabilistic inference will be discussed considering the limit on the amount of subject level data that can be acquired where privacy is concerned.

2.1 Missing Data

Lets assume V to be an indicator matrix showing if a variable is missing or observed, where $V_{ij} = 0$ if x_{ij} is missing, and $V_{ij} = 1$ if x_{ij} is observed. This thesis follows the convention where n represents the unobserved data ($V_{ij} = 0$) and m represents the observed data ($V_{ij} = 1$). Therefore, the data series p can be regenerated from n and m.

2.1.1 Missing Data Patterns

Patterns of missing data can be revealed in the data indicator matrix R. By organising the variables and cases with missing values in a dataset, the patterns in which missing data occurs can be derived (Song and Shepperd, 2007). Generally, two types of missing data patterns exists, namely; univariate and multivariate patterns. In the univariate pattern, missing values are present in only one variable. For example, only the the variable x_2 contains 3 missing values in table 2.1

On the other hand, more than one variable will contain missing values in the multivariate missing data pattern. This pattern can be further refined into: monotone and arbitrary patterns (Song and Shepperd, 2007). In the former, missingness occurs in such a way that in a set of variables $x_1, x_2, ..., x_n$, if the value in x_i is missing, then $x_{i+1}, ..., x_n$ will be missing too, as seen in table 2.2. In the arbitrary pattern, missingness can occur randomly over any variable with no special structure. An example can be seen in table 2.3. It is noteworthy that missing values are indicated by (?) in the below tables.

Table 2.1 Univariate Missing Pattern

	x_1	x_2	x_3	x_4	x_5	x_6
C1	*	*	*	*	*	*
C2	*	?	*	*	*	*
C3	*	?	*	*	*	*
C4	*	*	*	*	*	*
C5	*	?	*	*	*	*

2.1.2 Missing Data Categories

According to Marlin (2008), there are fundamentally two ways to categorise the distribution of data, i.e., the latent variables and response indicators. The factorization of the joint

Table 2.2 Monotone Missing Pattern

	x_1	x_2	x_3	x_4	x_5	x_6
C1	*	*	*	*	*	*
C2	*	*	*	*	*	?
C3	*	*	*	*	?	?
C4	*	*	*	?	?	?
C5	*	*	?	?	?	?

Table 2.3 Arbitrary Missing Pattern

	x_1	x_2	x_3	x_4	x_5	x_6
C1	*	?	*	*	*	*
C2	?	*	*	*	*	*
C3	*	*	*	*	*	?
C4	?	*	*	?	*	*
C5	*	*	?	?	?	*

distribution in the first case is presented as $P(O|X,Z,\mu)\ P(X,Z|\theta)$. $P(O|X,Z,\mu)$ represents the missing data model, while $P(X,Z|\theta)$ is the model with complete data. The intuition in the missing data model is that the probability of response depends on the true value of the latent variables and data vector in the distribution. This condition is reversed in the second factorization represented as $P(X,Z|O,\omega)\ P(O|w)$. Here, the intuition is that each missing data pattern specifies a different latent variables and data distribution (Vink, 2022). In this chapter and the remainder of the thesis, the author follows the intuition of the first factorization. However, it is important to note that the second factorization is equally a valid intuition.

The classification of missing data mechanisms is important as it aids the selection of suitable strategies for handling different types of missing data. According to Little and Rubin (2019), three important patterns of missing data exists, namely; missing at random

(MAR), missing completely at random (MCAR) and not missing at random (NMAR). These mechanisms help in explaining the reasons why some values may be unobserved or missing from a given distribution. A detailed description of missing data mechanisms is highlighted below.

Values are said to be missing at random (MAR) when the probability of a missing value on an attribute Y depends on the value of another attribute X but not on the value of Y itself. In other words, the probability that a value is missing, based on the MAR mechanism, is dependent on the relationship between the variable Y and the variable X and not the outcome value of the variable Y itself (Bashir, 2019). For example, consider a situation where the final mark of a student is missing, an assumption that the value is MAR applies when the missing final mark depends on the status of the student, but not on the student's final grade. Therefore, information about the students' status can be used to conveniently fill-in (predict) the missing final marks.

Lets assume X to be an n x p matrix containing incomplete data, where the columns p are the number of measured variables in the matrix and the rows, n, represents the sample size. The symbol X_{obs} represents the cases that have been observed and X_{miss} denotes that a value is missing . If another matrix R is considered, which spots the positions of missing values in X, the observations of X and R can be represented as x_{ij} and r_{ij}, respectively. Therefore, $r_{ij} = 1$ when x_{ij} is observed and $r_{ij} = 0$ when x_{ij} is missing. The distribution of R will therefore depend on $X = (X_{obs}, X_{miss})$. Thus, data is assumed to be MAR if:

$$Pr(R = 0|X_{obs}, X_{miss}, \Psi) = Pr(R = 0|Y_{obs}, \psi), \tag{2.1}$$

where ψ represents the parameters of missing values in the model. This indicates that the probability of missing values is dependent on the parameter estimates in the model.

$$Pr(R = 0|X_{obs}, X_{miss}, \Psi) \tag{2.2}$$

Data is said to be missing completely at random (MCAR) when the value of an attribute with missing data neither relies on the missing data nor the observed data. MCAR is a mechanism that is considered in most fields to be "totally and randomly" missing. Here, the probability of an attribute Y having missing values is not directly related to the output of the variable X or the output value of Y itself. For example, considering a dataset that contains the marks of students, an assumption that a student's final score is missing does not depend on the status of the student, neither does it depend on the grade of other students. The equation below denotes the MCAR mechanism. According to Little and Rubin (2019), this mechanism implies that all units have equal probability of being missing.

$$Pr(R = 0|X_{obs}, X_{miss}, \Psi) = Pr(R = 0|\psi) \tag{2.3}$$

Assuming we have an organised data distribution, when the probability of missing values on an attribute Y depends solely on Y itself, and does not depend on the value of another attribute X, data is said to be not missing at random (NMAR). In the same example considering a student's grade, if the same student's final score is missing, and its missingness depends on the final score itself (i.e. only scores within a particular range, say 50-60%, are missing), only final marks of other students can be substituted to impute these missing values. The NMAR mechanism is often considered to be more complex in statistical analysis. This can only be addressed by investigating reasons why participants in a study would choose to withhold some certain information, which is considered to be Y_{miss} itself (Alsaber et al., 2021). An illustration of missing data classification scheme by Little and Rubin (2019) can be seen in Table 2.4.

Not all missing data techniques are capable of handling categorical data. Therefore, less complicated missing data patterns were considered for imputing numeric data as all the benchmark techniques considered work effectively on numeric data.

Table 2.4 The proportion of petal length (X) to petal width (Y) on the iris dataset

	Y			
X	**Complete**	**MAR**	**MCAR**	**MNAR**
1.4	0.2	0.2	0.2	-
1.4	0.2	0.2	0.2	-
1.3	0.2	0.2	-	-
1.5	0.2	0.2	0.2	-
1.4	0.2	0.2	0.2	-
1.7	0.4	0.4	-	0.4
1.4	0.3	0.3	0.3	0.3
1.5	0.2	0.2	-	-
1.4	0.2	0.2	0.2	-
1.5	0.1	0.1	0.1	0.1
1.5	0.2	0.2	-	-
1.6	0.2	0.2	0.2	-
1.4	0.1	0.1	0.1	0.1
1.1	0.1	0.1	-	0.1
1.2	0.2	0.2	0.2	-
1.5	0.4	0.4	0.4	0.4
1.3	0.4	0.4	-	0.4
...	...	...	...	...
6.1	2.3	-	2.3	2.3
5.6	2.4	-	2.4	2.4
5.5	1.8	-	1.8	1.8
4.8	1.8	-	1.8	1.8
5.4	2.1	-		2.1
5.6	2.4	-	2.4	2.4
5.1	2.3	-	2.3	2.3
5.1	1.9	-	1.9	1.9
5.9	2.3	-	2.3	2.3
5.7	2.5	-	2.5	2.5
5.2	2.3	-	-	2.3
5	1.9	-	1.9	1.9
5.2	2	-	2	2
5.4	2.3	-	-	2.3
5.1	1.8	-	1.8	1.8

* Dashes indicate missing values

2.1.3 Missing at Random Assumption Based on Multivariate Data

The implication and interpretation of the missing at random assumption is quite straight forward when considering multivariate missing values with arbitrary patterns. From various examples stated by Little and Rubin Little (1992), an assumption is made that a data vector can be divided into distinct parts, where one part contains complete records and another is

subject to non-response. If the observed sub-vector is defined as $\mathbf{p}^a$, then the missing at random condition is satisfied by the missing data model $P(\mathbf{V} = \mathbf{v}|\mathbf{P}^a = \mathbf{p}^a, \mu)$.

When considering missing data with arbitrary patterns in multivariate settings, it is more intuitive to consider the missing at random condition as constraints imposed on the parameters of the given distribution $P(\mathbf{V} = \mathbf{v}|\mathbf{P} = \mathbf{p}, \mu)$. Lets recall that the definition of the missing at random condition states that when considering a prescribed value of v, $P(\mathbf{V} = \mathbf{v}|\mathbf{P} = \mathbf{p}\,', \mu)$ must also take the same value for the complete vector $p\,'$ that supports the observed measurement of $\mathbf{p}$ as determined by $\mathbf{v}$.

For cases where $\mathbf{p}$ is a vector of finite values, the conditional distribution can be parameterised as $P(\mathbf{V} = \mathbf{v}|\mathbf{P} = \mathbf{p}\,', \mu) = \mu_{vp\,'}$. Constraints are then specified on the parameters $\mu_{vp\,'}$ based on the missing at random assumption. Specifically, if complete vectors $\mathbf{w}$ and $\mathbf{u}$ both support the observed dimensions of $\mathbf{p}$ as specified by $\mathbf{v}$, the condition for MAR will require that $\mu_{vw} = \mu_{vu}$.

Table 2.5 Restrictions imposed on $P(V = v|P = p)$ parameterisation based on the MAR assumption

P/V	0 0	0 1	1 0	1 1
0 0	α	β	γ	$1-\alpha-\beta-\gamma$
0 1	α	σ	γ	$1-\alpha-\sigma-\gamma$
1 0	α	β	Γ	$1-\alpha-\beta-\Gamma$
1 1	α	σ	Γ	$1-\alpha-\sigma-\Gamma$

It is noteworthy that the missing at random condition applies to both the individual response vectors $\mathbf{v}$ and the corresponding data vectors $\mathbf{p}$. It is possible to experience cases where missing data generated in a particular data case may be Missing at Random, while the missing data observed in another case may be Not Missing at Random. For a missing data model to only generate data that is missing at random, the parameter constraints for the MAR condition must apply to every vector $\mathbf{v}$ that corresponds to the instantiation of $\mathbf{p}^o$.

To further explain the restrictions imposed on the parameterisation of $P(V = v|P = p)$ based on the MAR assumption, lets expand on an example that was presented by (Little and Rubin, 2019). Considering two vectors in a two dimensional binary data that is subject to non-response on both dimensions, the MAR assumption imposes a set of constraints on the parameter values of $P(V = v|P = p, \mu)$ which requires that certain elements of the conditional distribution are equal.

As observed from table 2.5, when **V** = [0, 0], neither P_1 nor P_2 is observed and the MAR condition requires that $P(V = [0,0]|P = p)$ must be equal for all values of **p**. In the case where **V** = [0, 1], P_2 is observed. Here, we get the restriction that $P(V = [0,1]|P_1 = p_1, P_2 = 0)$ and $P(V = [0,1]|P_1 = p_1, P_2 = 1)$ must be equal for all values of p_1. When **V** = [1, 0], P_1 is observed and a restriction is placed where $P(V = [1,0]|P_1 = 0, P_2 = p_2)$ and $P(V = [1,0]|P_1 = 1, P_2 = p_2)$ must be equal for all p_2. The missing at random assumption places no restrictions on $P(V = [1,1]|P = p)$ when all the dimensions are complete. Its value will simply be determined using the normalisation constraint.

The significance of the missing at random condition is that it imposes some uniformity on the model with missing data, which increases the chances of determining $P(V = v|P = p, \mu)$ when only **v** and the observed values of **p** are given. For example, in a case where $p_2 = 0$ and p_1 is unobserved, so that $v = [0,1]$, despite the fact that the value of p_1 is unknown, we can still ascertain that the value of $P(V = v|P = p, \mu)$ must be β when the $p_1 = 0$ and $p_1 = 1$. After the MAR condition has been satisfied, the values present in **p** will always contain the exact information required to determine $P(V = v|P = p, \mu)$.

From a modelling point of view, it is more sensible to assert that a random variable V_1 is either dependent on a random variable P_1, or is independent of the random variable P_1. If the variable V_1 always depends on P_1 and the value of P_1 is missing in a given instance, then P_1 is not missing at random. This mechanism will make parameter estimation more challenging. On the other hand, the condition for MAR will allow V_1 to be dependent on P_1 for only some

values of V_1. This condition is convenient, but yet difficult to justify. Research carried out by Little and Rubin Little (1992) acknowledges that assuming a MAR condition for arbitrary missing data patterns in a multivariate setting is not always realistic, but sometimes produce reasonable results if a sufficient amount of covariates are observed to produce accurate response patterns.

2.1.4 Effect of Missing Data on Inference

Research carried out by Rubin (1976) assessed the impact of missing values on inference drawn from Bayesian analysis. Findings from this study were also represented in more recent studies by (Schafer, 1997; Marlin, 2008). In this section, a description of missing data effects is presented considering random missing mechanisms which is experimentally demonstrated in the later chapters of this thesis.

Considering a linked parametric model with response indicators, latent and data variables which forms $R(D|P,V,\mu)R(P,V|\theta)$, and assumes the factorised posterior distribution $R(\theta|\sigma)R(\mu|\omega)$. Given a range of missing data vectors p_n where $n = 1,\ldots,N$ for the posterior data distribution $R(\theta|\{p_n,d_n\}_{1:N,\sigma,\omega})$ on θ, the equation can be represented as:

$$R(\theta|\{p_n,d_n\}_{1:N,\sigma,\omega}) \propto R(\theta|\sigma)\int R(\mu|\omega)\prod_{n=1}^{N}\int\int R(p_n^o,p^m,v|\theta)R(d_n|p_n^o,p^m,v,\mu)dp^m dv d\mu \tag{2.4}$$

Without simplification on the assumptions, the complete data and missing data models are joined together through integration over both the latent and missing values. Under MCAR and MAR conditions, $R(d_n|p_n^o,p^m,v,\mu)$ remains constant for all the values of p^m and v when d_n and p_n^o is fixed. By implication, the posterior values of θ are independent from d_n,μ and ω. Therefore, the model for the missing data can be excluded from the posterior. Integration

over the missing values will then be insignificant within the model of the complete data. As a result, this simplification generates the observed posterior given as:

$$R^{obs}(\theta|\{p_n,d_n\}_{1:N,\sigma}) \propto R(\theta|\sigma)\prod_{n=1}^{N}\int R(p_n^o,v|\theta)dv \tag{2.5}$$

Table 2.6 Specification of $R(P=p)$ and $R(D=d|P=p)$ which satisfies the MAR condition.

p	$R(p)$	$R(D=[0,0]\|p)$	$R(D=[0,1]\|p)$	$R(D=[1,0]\|p)$	$R(D=[1,1]\|p)$
0 0	a	α	β	γ	$1-\alpha-\beta-\gamma$
0 1	b	α	σ	γ	$1-\alpha-\sigma-\gamma$
1 0	c	α	β	Γ	$1-\alpha-\beta-\Gamma$
1 1	d	α	σ	Γ	$1-\alpha-\sigma-\Gamma$

When the MAR and MCAR conditions stand, the missing data model will not have effects on the validity of inference drawn from the data. However, when the MAR and MCAR conditions fail, data is said to be Not Missing at Random (NMAR). Therefore, basing the inference of θ on the observed data and neglecting the missing data model will lead to biased predictions and inference on the model parameters.

It is noteworthy that in cases where the MAR condition does not hold, using an arbitrary missing data model will not be sufficient as this will lead to the generation of biased estimates. According to Marlin (2008), a more astonishing observation is that in an instance where the MAR assumption stands in a training model, the inference drawn from parameters in a simpler model may still be biased. This is a challenge in machine learning tasks where models are often applied to produce attributable learning outcomes.

To further illustrate the aforementioned issue, lets assume a simple binary data setting with vectors **p** subject to missing data on two dimensions. The data distribution $R(P=p)$ as well as the missing data mechanism $R(D=d|P=p)$ is represented in Table 2.6. Note that the model satisfies the MAR condition that is described in Section 2.1.3.

Since a detailed consideration of the processes that cause missing data in respect to their impact on inference will require robust models for experimental analysis, previous research has shown that making proper likelihood and Bayesian inference is more straight forward for many cases (Rubin, 1976; Little and Rubin, 2019).

2.2 Privacy Preserving Data Mining

The wide acceptance of emerging technologies have challenged and disrupted the knowledge of information privacy. Approaches to privacy from a legislative perspective date as far back as 1948 when the Universal Declaration of Human Rights (UDHR) was adopted by the United Nations (UN), which states in Article 12 that "No one shall be subjected to arbitrary inference with his privacy, family, home or correspondence, nor to attacks upon his honour and reputation" (UN, 1948).

The paradigm of private and secure computations in distributed systems present cryptographic solutions to the issues of privacy and security of computations (Jagannathan and Wright, 2008). The notion of privacy in such a setting is defined by a comparative evaluation of the information learned by parties towards completing a distributed protocol to the information such parties would learn if computations on the distributed protocol were completed for them by a trusted third party (TTP).

According to Agrawal and Srikant (2000), Privacy-preserving Data Mining (PPDM) focuses on secure multiparty computation of data for the purpose of establishing a standard level of confidentiality. However, the level of confidentiality established is not customised to the personal needs or context-relevant requirements of the individuals described by such information. According to research, the context of PPDM can be considered in respect to specific privacy theories (Jagannathan and Wright, 2008). In the following sections, PPDM related theories applicable to the framework proposed in section 6.5 are presented.

2.2.1 Restricted Access Theory

This theory is premised on the knowledge of how personal information (PI) can spread after it has been disclosed (Gavison, 1980). The proliferation of personal information can be described below:

(a) At the collection stage, an individual discloses some level of information about themselves to a third party (or third parties).

(b) A third party or third parties now posses a copy (or copies) of the information shared from the source.

(c) The proliferation of the individual's information may now be possible as a third party (or third parties) now have information that can be shared with other parties.

The growth of information technology (IT) usage has exacerbated this challenge by enabling quick and easy sharing of information. Therefore, research has identified the need for regulatory frameworks to tackle this issue (Wahlstrom and Fairweather, 2013).

Research carried out by Elgesem (1999), viewed restricted access as a characteristic of privacy. The author further opined that unfamiliar individual's in the world have restricted access to ones personal information. Thus, this theory is pervasive and is difficult to distinguish between private and public contexts. Tavani and Moor (2001), argued that the above theory only points to an individual's right to privacy (normative) rather than observable scenarios where privacy is concerned (descriptive). For instance, the privacy that an individual has may or may not be observable, but this does not enhance nor detract their right to privacy. In addition, Elgesem (1999) stated that an individual's PI can be disclosed privately. However, based on the restricted access theory, there is loss of privacy in every scenario where personal information is disclosed. For instance, when a person A confides in person B and person B treats the information shared with some level of confidentiality, then there is no risk associated. In addition, the consent of person A is implicit. Therefore, their consent (control)

is applicable to the restricted access theory. A highlight of the dissimilarities between privacy and privacy management was presented by Tavani and Moor (2001) in response to the above. The author identified consent as an important step in privacy management and states that an individuals right to privacy is not enhanced or detracted whether the individual does or does not consent to the proliferation of their PI.

According to Allmer (2011), many circumstances occur where individuals can restrict access to their PI, thus making regulatory rules in such circumstances to be redundant. Furthermore, Martin (2012) argued that the restricted access theory is conflicting as it states that information can either be accessible or inaccessible. By reapplying the distinguishing factors between normative and descriptive view to privacy in the works of Tavani and Moor (2001), a combination of privacy and data security can be seen in the restricted access theory.

This theory is consistent with privacy preserving data mining (PPDM) as it ensures secure multi-party computation therefore restricting the ability of third parties to access or extract information from any data that is shared.

2.2.2 Control Theory

This theory was pioneered by Westin (1970) and views privacy as the need for an individual to have the ability to control how, when and to what extent their information is being shared with other parties. This theory stipulates that where an individual can effectively control who has access to their personal information, then they have privacy.

In this theory, control over one's information is confined on the initial disclosure of that information (Elgesem, 1999). Once information has been disclosed to a third party, an individual no longer has control over its proliferation, thus, a loss of control (loss of privacy) has been incurred. This theory is true even if a third party does not view or disclose the information that has been shared. However, in such conditions, no observable loss can be

seen in the privacy of the individual who owns such information, neither is any risk associated (Elgesem, 1999).

This theory however may not be applicable to all aspects of PPDM as provisions for informed consent are not considered in PPDM. It is however relevant to the extent that consent is sought for before implementing PPDM.

Chapter 3

Missing Data Handling

3.1 Introduction

This chapter reviews state-of-the-art approaches to handling the challenges of missing data. The significant increase in the sensing capabilities of IoT devices has raised new challenges in data acquisition and processing. For example, sensor devices deployed over large geographical areas produce high-dimensional spatial and temporal data (Barnaghi et al., 2015). In practice, part of these sensor readings may be lost at the data acquisition stage owing to issues relating to data storage, sensor failure or lost communication between sensor nodes (Genes et al., (2016); Agbo et al., (2020)). Algorithms and techniques used in data mining processes generally do not produce good analysis results from incomplete datasets and this can have huge implications on the inference made from any given dataset (Little and Rubin, 2019).

3.2 Missing Data Treatments

Missing data research has gained popularity in diverse fields and consequently, numerous methods have been proposed in literature to handle this problem. Generally, missing data

techniques can be grouped into two main categories: traditional techniques and modern techniques (Raja and Thangavel, 2020). Traditional techniques are generally easy to implement whereas, modern techniques require much powerful hardware and software resources (Bashir, 2019).

Numerous missing data techniques have been presented in literature. In this chapter, the author presents some of the most popular missing data handling techniques.

3.2.1 Deletion Methods

A popular technique for handling missing data in early research is the complete case analysis (CCA) approach, otherwise known as listwise deletion. This method simply discards an entire record of observations if one or more instances are missing. Various missing data problems are treated using this method as it is fairly straight forward and easy to implement. However, to reduce the level of bias from the application of this technique, it is important to consider the mechanism in which missing data occurs (Baraldi and Enders, 2010). For instance, if data is MCAR, an unbiased outcome will be produced from the analysis. In addition, this technique only performs well when only few values are missing from a record of observations and does not perform well when a large number of values are missing (Bashir, 2019). After applying CCA technique, the data analysis process will make use of only the cases that have all observations present.

Another popular deletion technique is the pairwise deletion approach. Here, observations with missing values are ignored at different stages of the analysis process. That is, some observed cases may be selected for some specific analysis but may not be used for all analyses. This technique is also known as available case analysis (ACA) (Bashir, 2019). This method generally produces better results than the CCA technique because it reduces the amount of data that is ignored from a record of observations. Similar to CCA, ACA technique only performs better when data is MCAR (Peugh and Enders, 2004).

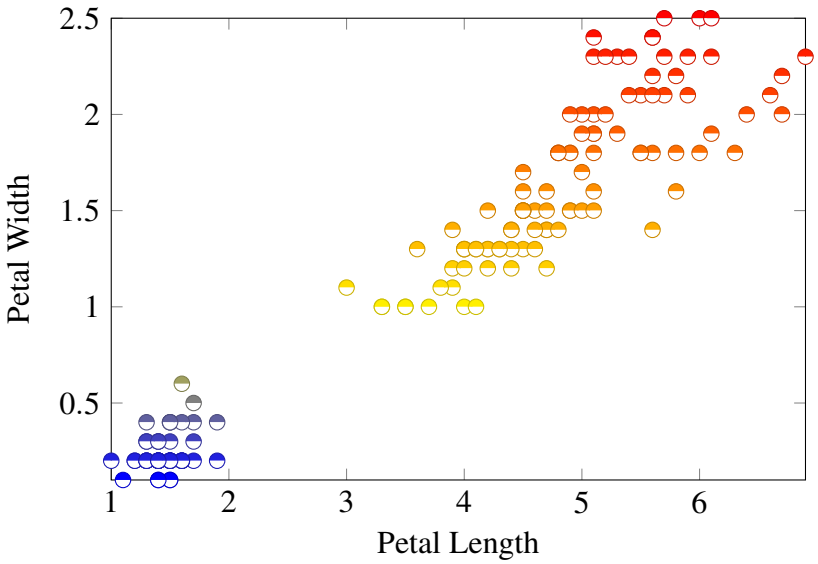

Fig. 3.1 Complete data

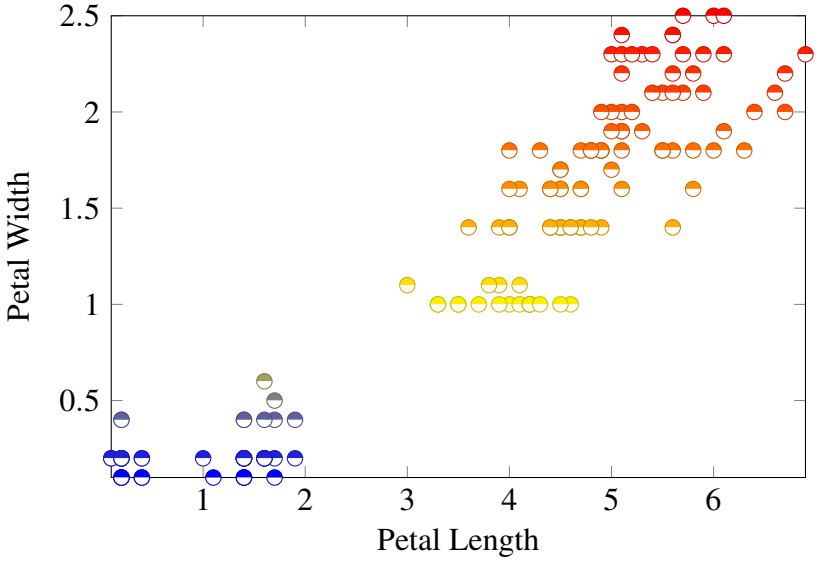

Fig. 3.2 Incomplete data

To throw more light on the principles behind the deletion approaches, lets consider the data presented in Figure 3.1. The positive correlation between X and Y (0.969) shows a steady relationship between petal length and width. The scatter plot in Figure 3.2 shows the outcome of the deletion approaches where data is MAR.

As deletion techniques will only make use of cases with fully observed values of Y, it ignores values starting from where petal length is equal to 5.6 on wards (see Table 2.4). The estimated mean value of Y in the complete dataset is 1.198, whereas after applying deletion approaches, the estimated mean value becomes 1.0736. Consequently, the mean value of X (3.758) will become 3.494 after applying deletion methods. When considering the standard deviation, the value of Y (petal width) has a standard deviation of 0.763 for the complete dataset, in contrast to the deletion methods which has a standard deviation value of 0.726.

In order to retain the predictive power of a dataset, it is necessary to use the available observations to estimate and replace missing values (Genes, 2018). This process is known as imputation. The following section discusses popular traditional imputation techniques for filling in missing values.

3.2.2 Mean Imputation

This method is the most straightforward among other imputation methods. Here, missing values are filled in using the mean value of the observed data, assuming that data is MCAR (Enders, (2006); Wilks, (1932)). Although this method is beneficial in providing a quick solution to the missing data problem, it does not show the variability of records in a given sample or present any knowledge from the results (Mohamed et al., 2018). Lets assume that x_4 in Table 2.3 is a continuous variable. The missing values in C4x_4 and C5x_4 will be replaced with the mean value of the three observed variables C1x_4 to C3x_4. This can also be represented as

$$\bar{x}_4 = \frac{1}{3}\sum_{i=1}^{3} C_i(x_4) \tag{3.1}$$

If x_4 contains a categorical variable, the missing values in C4x_4 and C5x_4 will be calculated using the mode of the three observed variables in x_4 such that:

$$\bar{x}_4 = max[\frac{|C_i(x_4|C_i(x_4) \equiv v)|}{3}] \tag{3.2}$$

where v is a known value for the variable x_4.

3.2.3 Regression Imputation

This is another traditional technique that imputes missing values using the observed training data of the output variable Y (Song and Shepperd, 2007). Generally, regression imputation involves two phases: first, available data is used to build a regression model and secondly, missing values are imputed using the predicted outcome from the regression model (Song and Shepperd, 2007).

According to Little (1992), regression imputation of missing values can be approached using independent variables alone or both dependent and independent variables. In the former, the independent variables in a given case are used to impute other independent variables that are missing. The author suggested the use of Weighted Least Squares (WLS) regression to perform this imputation. However, Gourieroux and Monfort (1981) argued that the estimation error in the WLS method results in a correlation among incomplete variables which affects the weight selection and the consistency of the standard estimation error. The authors further proposed the use of Generalised Least Squares (GLS) regression to solve this problem. GLS regression uses both dependent and independent variables to impute missing values considering that the correlation between the dependent variable and a missing independent variable is high. Further investigations carried out by Little (1992) noted that the

estimated standard error of the regression coefficients on imputed data derived from WLS or Ordinary least squares (OLS) will tend to be small whether the missing independent variable is filled in using independent variables alone or both dependent and independent variables.

Generally, in regression imputation, a regression model is used to approximate missing features by using data that is available. In the first instance, a regression equation will be used to predict values of incomplete instances using complete cases. These predicted values are then generated for the incomplete variable and used to predict and fill in missing instances (Emmanuel et al., 2021). To impute missing instances $(y_1, \cdots, y_q)$ for the output variable Y, this thesis considers the following regression model:

$$Y = \alpha + \beta_1 y_1 + \cdots + \beta_q y_q + \varepsilon \tag{3.3}$$

where $\alpha, \beta_1, \cdots, \beta_q$ are unknown variables and ε represents a distance variable. The estimates produced in equation 3.3 will yield a prediction for Y given the variables:

$$\hat{Y} = a + b_1 y_1 + \cdots + b_q y_{q,i} \tag{3.4}$$

where a, b_1, b_q represent the lease squares estimates for $\alpha, \beta_1, \cdots, \beta_q$. An imputation $\tilde{Y}$ is further made for each instance with missing values:

$$\tilde{Y} = \hat{Y} = a + b_1 y_{1i} + \cdots + b_q y_{qi} \tag{3.5}$$

The evolution of modern techniques for handling missing data can be attributed to the works of (Little and Rubin, 1987). Today, various research has been carried out which provides a good background on modern missing data imputation methods (Graham, (2009); Blackwell et al., (2017)). For the purpose of this research, a highlight of some modern imputation methods is discussed, which will be revisited in the later chapters of this thesis.

3.2.4 Multiple Imputation (MI)

Research carried out by Marshall et al. (2010) showed the merit of MI for imputing missing values, especially when the rate of missing data is above 10%. An advantage of multiple imputation over single imputation is that single imputation methods tend to underestimate the variance that may exist in a given distribution in some cases. Therefore, MI methods have been proposed to overcome this limitation (Little and Rubin, 2019).

Multiple imputation method has shown efficiency in obtaining quality results in addition to the ease of implementing this technique using advanced software and programming (Alsaber et al., 2021). In the imputation process, random errors are identified, which helps in obtaining unbiased estimates for every given parameter with missing values. MI technique also provides results that are suitable with smaller sample sizes or when the ratio of missing data is large, while departing from normality assumptions (Marshall et al., 2010).

According to Allison (2000), some factors are important in order to obtain good results from the MI algorithm. Firstly, missing data should be missing at random (MAR). Secondly, the method used to impute missing values should be suitable for subsequent analysis. This is important for maintaining the association among variables. Thirdly, the imputation model should match with the model for the overall investigation.

The application of multiple imputation has been seen in various domains such as environmental studies (Kotsiantis et al., 2006), healthcare research (Chang et al., 2020), industrial data bases (Lakshminarayan et al., 1999) and survey data (Van Ginkel et al., 2007). Three stages are involved in MI namely; Imputation, Analysis and Pooling stage. A brief description these stages are outlined below.

Imputation Phase

In this first phase, different iterative algorithms can be applied. However, the augmentation approach is the most relevant where data is normally distributed (Baraldi and Enders, 2010).

The imputation phase is further divided into two smaller processes: imputation and posterior procedure.

The **imputation procedure** generates different data sets with diverse missing data predictions. According to Schafer and Graham (2002), the number of data sets generated varies from 15 to 20. This is similar to an augmentation algorithm, likewise a stochastic imputation approach in that it makes use of a mean vector and covariance matrix to generate a regression model. The estimated values in these models are used to replace missing values. Often times, the residual values with a constant variance and zero mean are included in the newly imputed values.

The next sub-process involved in the imputation phase is the **posterior procedure**. This step relies on Bayesian method for the estimation of regression model parameters (covariance matrix of the estimated values and the mean vector). Semantically, this step calculates the estimates of the parameters in the data imputed from the imputation step and further appends a residual variation on each estimated values. Next, a new set of parameters are generated, which differ from the parameters used for missing data imputation in the previous step. A new data set is further produced by using the imputation estimates produced by the new regression models generated from the new covariance matrix and mean vector obtained in the previous posterior step. The new dataset will contain values which differ from the values produced in the previous imputation step. Iterating these two sub-techniques in the imputation phase up to 100 times will produce multiple copies of the data set (Little and Rubin, 2019).

The main goal of the first phase is to generate different copies of a dataset with each data set having values that differ from one another. A random error value is added to each imputed case which results in the variation that exists between each generated dataset. An auto-correlation exists between the imputation and posterior procedures, causing the overall imputation phase to be challenging when large amounts of data are missing. For instance,

assuming an imputation phase that requires the generation of 15 data sets, if the imputation and posterior procedures iterate about 200 times, the entire imputation phase will need to iterate about 3000 times, thereby taking much time to complete. This challenge makes the multiple imputation technique less practical for very large data sets with large amounts of missing values (Peugh and Enders, 2004).

Analysis Phase

After the required number of data sets have been generated, each data set is analysed using statistical methods in the analysis phase. This phase is the most straightforward among the MI phases. The main aim of this phase is to prepare the data sets generated from the imputation phase for the next stage (Enders, 2010).

Pooling Phase

This phase combines the average and standard errors of parameter estimates in a single dataset. Research has shown that the standard errors and average parameter estimates can be calculated using statistical formula and three basic statistical equations are employed in this phase (Rubin, 1996):

- Computing the average of the squared standard errors in each generated data set. Given by;

$$\bar{E} = \frac{1}{p}\sum_{k=1}^{p}\widehat{E}_k \tag{3.6}$$

  where $\widehat{E}_k$ represents the squared standard errors of the k^{th} generated data set and p is the total number of data sets generated in the imputation phase.

- Measuring the variance between parameters in the imputed data sets.

$$\sigma_\rho = \frac{1}{p-1}\sum_{k=1}^{p}(\widehat{\rho}_k - \bar{\rho})^2 \tag{3.7}$$

where σ_ρ represents the variance of parameters, $\widehat{\rho}_k$ represents the parameter estimate for the k^{th} generated data set and $\bar{\rho}$ is the mean value of the parameters in the system.

- Calculating the overall standard error in the distribution.

$$SE = \sqrt{\bar{E} + \sigma_\rho / p} \tag{3.8}$$

Although the stages involved in this approach are tedious, more robust systems and software packages have been developed to perform MI effectively (Bashir, 2019).

3.2.5 Expectation Maximization Algorithm

Research carried out by Anderson (1957) introduced the basic concept of the maximum likelihood function and further explained it using simple steps. If we have two systems A and B, with the same input data X. Lets assume that system A has all its output data Y complete and the output data in system B are all missing. To estimate the missing output data in system B, we first of all calculate the variance and average for the input variable X on both systems and uses the output data Y on system A to determine a linear estimate for the output parameters of system B. These estimated parameters can further be used to predict every missing value in system B. This research by Anderson (1957) was based on an assumption that missing data occurs in a monotone pattern (see 2.1.1). However, in an ideal system, these steps will need to be carried out iteratively (Schafer and Graham, 2002). To address this, the "Expectation maximization" (EM) algorithm was proposed to provide a suitable solution to the prevalent missing data problem (Dempster et al., 1977).

The general idea behind the EM algorithm is to iteratively estimate the parameters required to predict the missing values in a given distribution by calculating the mean and variance between parameters (Bashir, 2019). The application of the EM algorithm usually aims at solving missing data problems. However, research has also shown the use of this

algorithm in solving complex problems for complete data sets. For example, multilevel linear models, structural equation model and finite mixture (Liang and Bentler, (2004); Muthén and Shedden, (1999); Neale et al., (1999); Agbo et al., (2022)). The following section describes the use of the EM algorithm for estimating a linear regression model based on variance and mean vector.

The EM algorithm is an iterative process which consists of two steps: the expectation step ("E-step") and maximization step ("M-step"). The initial values available in a system are required to initiate the estimation process. The E-step begins with the construction of a linear regression model, using the co-variance matrix and initial mean vector to produce estimates for the missing values from the observed data. The M-step follows after the E-step and produces new parameter values for the data that have been estimated. The algorithm saves the last co-variance matrix and mean vector to determine the next E-step and builds a new regression model from the results, which is then used to determine new estimates for missing values. The M-step will subsequently run again to determine new parameters from the updated estimates. The EM algorithm will iterate these two steps until the values of the co-variance matrix and mean vector no longer changes or converges, where the converged value corresponds to that of the value of the maximum likelihood estimates (Neal and Hinton, (1998); Agbo et al., (2022)).

To explain the mechanism of the EM algorithm, this thesis considers a similar data case for the analysis of a single variable, where X represents the input variable with complete data and Y will be the output with incomplete data. Lets consider few data points with single input/output variables to provide a simplified description of the system. The following equations are applied to establish the parameters using the maximum likelihood approach, for the case of missing data (Enders, 2010).

$$\mu_U = \frac{1}{N}\sum_{i=1}^{N} U_i \tag{3.9}$$

$$\sigma_U^2 = \frac{1}{N}\left(\sum_{i=1}^{N} U_i^2 - \frac{(\sum_{i=1}^{N} U_i)^2}{N}\right) \tag{3.10}$$

$$\mu_Y = \frac{1}{N}\sum_{i=1}^{n} Y_i \tag{3.11}$$

$$\sigma_Y^2 = \frac{1}{N}\left(\sum_{i=1}^{N} Y_i^2 - \frac{(\sum_{i=1}^{N} Y_i)^2}{N}\right) \tag{3.12}$$

$$\sigma_{U,Y} = \frac{1}{N}\left(\sum_{i=1}^{N} U_i Y_i - \frac{\sum_{i=1}^{N} U_i \sum_{i=1}^{N} Y_i}{N}\right) \tag{3.13}$$

These equations reflect a significant aspect of the expectation step as they represent basic data points that are used to determine model parameters. I.e., the average of input and output data ($\sum_{i=1}^{N} U_i$ and $\sum_{i=1}^{N} Y_i$), the squared sum of the input and output data ($\sum_{i=1}^{N} U_i^2$ and $\sum_{i=1}^{N} Y_i^2$) and finally, the product of the input and output data ($\sum_{i=1}^{N} U_i \sum_{i=1}^{N} Y_i$) (Enders, 2010).

In the first instance, the E-step uses the initial conditions to determine estimates for missing values. After that, the M-step takes over and the imputed values will be substituted in equations (3.9) to (3.13) to determine new parameter estimates. The E-step then uses these new parameter estimates to build a linear model which will use the observed values in the system to predict missing values. Considering the system with single variable data, where missing values are present in the output Y, a linear model can be built using the following formula:

$$\hat{Y} = \beta_0 + \beta_1 U \tag{3.14}$$

$$\beta_1 = \frac{\sigma_{U,Y}}{\sigma_U^2} \tag{3.15}$$

$$\beta_0 = \mu_Y - \beta_1 \mu_U \tag{3.16}$$

$$\sigma_{U,Y}^2 = \sigma_Y^2 - \beta_1^2 \sigma_U^2 \tag{3.17}$$

Equation (3.14) represents a simple linear regression model, where β_0 and β_1 are the linear coefficients, $\hat{Y}$ is the predicted outcome and $\sigma_{U,Y}^2$ represents the variance between the input variable U and the output Y as shown in equation 3.17.

The imputation procedure for missing data is slightly complex due to the unavailability of sufficient statistics. However, this challenge is overcome in the expectation step by using the observed data to formulate initial conditions required to calculate sufficient statistics (Dempster et al., 1977). As a matter of fact, the borrowing of information from observed data for the purpose of predicting missing values is fully required by the EM algorithm. This process is called conditional expectation (Bashir, 2019). Subject to the mean value of the output data $\sum_{i=1}^{N} Y_i$ and the product of the input and output data $\sum_{i=1}^{N} U_i \sum_{i=1}^{N} Y_i$, the values of the predicted output is determined by equation (3.14). The E-step then computes sufficient statistics using the predicted values. An adjustment is made to the squared sum of the output data where:

$$\sum_{i=1}^{N} Y_i^2 = \sum_{i=1}^{N} \left(\hat{Y}_i^2 + \sigma_{U,Y}^2 \right) \tag{3.18}$$

where $\hat{Y}_i^2$ is the squared predicted output data. The E-step further substitutes the value of the squared sum of the output data in equation (3.12) with the result obtained in equation (3.18).

3.2.6 Imputation Based on K-nearest Neighbour

The KNN method is a machine learning algorithm which approaches imputation by classifying the closest neighbours to missing values and using these neighbours to impute missing values based on a distance measure between points (Maillo et al., 2017). KNN imputation can be carried out using various distance factors, such as the Manhattan distance, Minkowski

distance, Cosine distance, Hamming distance, Jaccard distance and Euclidean distance. However, research has shown that efficiency and productivity can be maximized by using the Euclidean distance, and this is the most widely used measure in the research community today (Amirteimoori and Kordrostami, (2010); Emmanuel et al., 2021). The KNN imputation technique can further be described using the Euclidean distance measure as represented below:

$$Dist_{xy} = \sqrt{\sum_{k=1}^{m} (X_{ik} - Y_{jk})^2} \tag{3.19}$$

where $Dist_{xy}$ represents the Euclidean distance, k represents the attributes of the data $j = 1,2,3\ldots k$, data dimensions, X_{ik} is the the attribute containing missing data and $_{jk}$ represents the value of j containing complete data. A weight mean estimate is calculated for the value of k points which have minimum distance as follows:

$$X_k = \frac{\sum_{j=1}^{J} w_j v_j}{\sum_{j=1}^{J} w_j} \tag{3.20}$$

where X_k represents the mean estimate, J represents the number of parameters that are used with $j = 1,2,3\ldots K$. The complete values on attributes that contain missing values are represented by v_j and w_j represents the nearest observed neighbours. The weighted value is further derived as follows:

$$w_j = \frac{1}{dis_{(x,y)^2}} \tag{3.21}$$

3.3 Statistical Description of Imputation Techniques

In order to demonstrate and compare the performance of the techniques for missing data treatment considering different mechanisms and missing data rates, the author simulated a stationary missing data series. Although the simulation results obtained from the data may

be challenging to extrapolate for real empirical data, the techniques still show tremendous advantages for real world missing data treatment.

By simulating the data collection process, we have full control over the data distribution, relevant covariates and the true parameters of the generated model are also known. This allows us to experimentally control the data compositions in this research, making it possible to investigate the strength of missing data treatment techniques under varying conditions. By knowing the true complete data, the researcher will be able to evaluate the accuracy of the treatment technique and control the type of missing data and mechanism.

3.3.1 Missing Data Formation

Missing data was simulated at 10% based on the MAR assumption to demonstrate the statistical description of some of the identified imputation techniques. All missing values in this case were generated on a single variable. The missing data mechanisms are connected to the covariate and the probability of a missing actor was set to a single ratio. This procedure was used to ensure that the missingness imposed on the dataset clearly reflected the desired mechanism.

3.3.2 Imputation Model Statistics

The analysis also exposed the proximity of the imputed model parameters to the complete data and also showed how well the inferential conclusions relate to that of the complete dataset. In this section, the author describes in detail, the processes involved in the earlier mentioned imputation techniques and further assess their efficiency in recovering missing values in a stationary dataset.

Considering the missing data case in Table 2.4 where data is MAR, the expected mean value of the output Y is 1.0736. This resulting mean value is then substituted for all missing records. Figure 3.3 shows a scatterplot of the imputed data through mean substitution at

1.0736 horizontally linear with zero slope on the Y-axis. In this instance, the input X and output Y has zero correlation as the missing data estimate depends on Y alone. Considering the outcome of the mean imputation technique, the correlation value between X and the newly imputed data $\hat{Y}$ is 0.89365 as opposed to 0.965 which is the correlation value of the complete dataset.

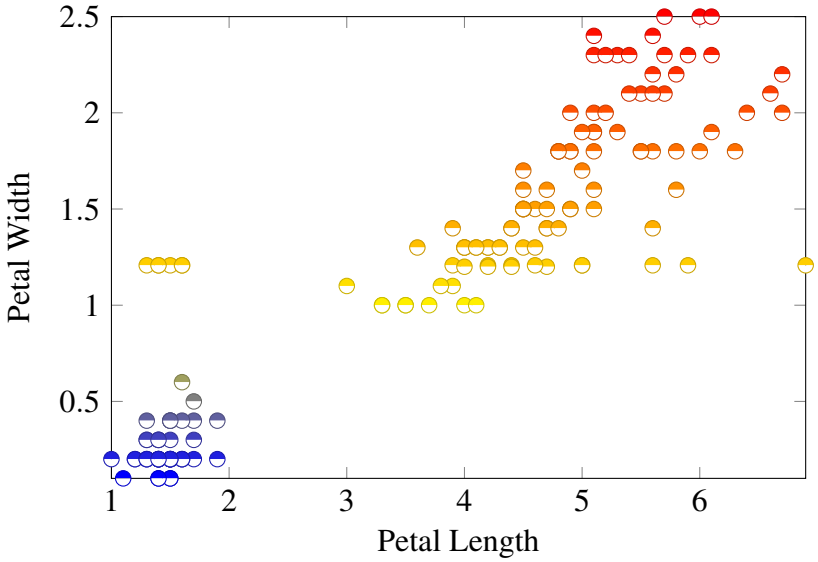

Fig. 3.3 Mean Imputation

The regression imputation process can also be described using the same mechanism as the previous. Here, the observed values of the output variable Y were used together with the values of the observed input variable X to fill in cases with missing values in the output variable Y using a regression model with the coefficient: $\hat{Y} = 0.404x - 0.3448$. Using the input X (observed data) on the regression model will produce the estimated output ($\hat{Y}$) which is finally used to estimate and fill in the missing values of Y. This method may however generate biased estimations as missing values are imputed using a linear model as seen in Figure 3.4. The values imputed in this method were generated from a linear function therefore, yielding a correlation value of 0.974 between the input X and output Y as opposed to 0.969 produced in the original data. Consequently, the variability of the imputed data will be reduced as a result of the imputation process. For instance, the original complete data

produced a standard deviation value of 0.763 for the output Y as opposed to 0.731 produced by the linear regression model. According to Graham (2009), the imputation of missing values using linear regression imputation and mean imputation will yield biased estimates when correlating the standard deviation of MAR and MCAR mechanisms.

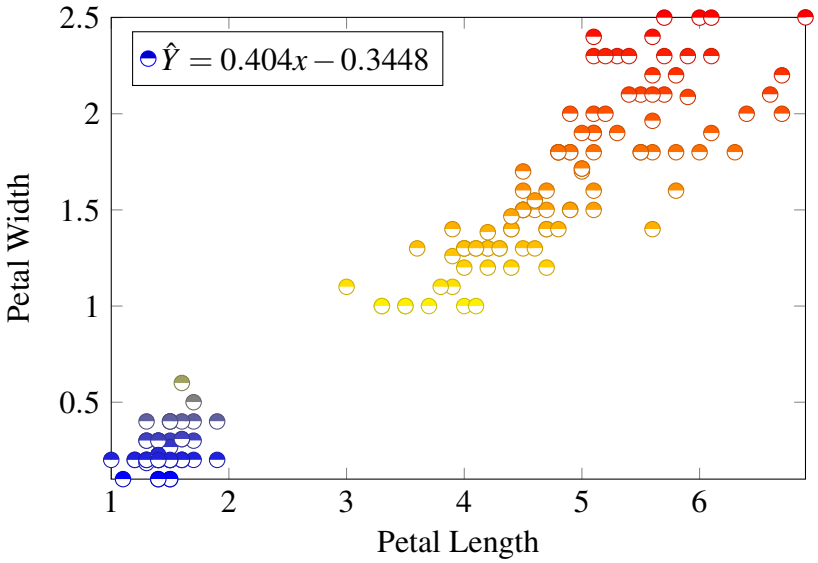

Fig. 3.4 Regression Imputation

To further explain the EM imputation model, lets take U to be the petal length and Y as petal width from the data in table 2.4. In the first phase of the EM algorithm, the author computes values for the initial model parameters, covariance matrix and mean vector. The initial values of the model parameters can also be computed using other simpler approaches such as regression imputation (Enders, 2010). In this example, the author performs CCA in order to obtain the initial parameter values represented as follows:

$$\mu_0 = \begin{bmatrix} \mu_U \\ \mu_Y \end{bmatrix} = \begin{bmatrix} 3.758 \\ 1.073 \end{bmatrix}$$

$$\Sigma_0 = \begin{bmatrix} \sigma_U^2 & \sigma_{U,Y} \\ \sigma_{Y,U} & \sigma_Y^2 \end{bmatrix} = \begin{bmatrix} 3.0924 & 0 \\ 0 & 0.5237 \end{bmatrix}$$

The algorithm constructs a linear model using the initial values of the parameter vectors and matrix in the first iteration. The model further imputes the missing values in the output variable (petal width) by making use of the complete data observed in the input variable (petal length). By substituting the parameter values from the mean vector μ_0 and the covariance matrix $\sum_0$, the following parameter values will be obtained:

$$\beta_1 = \frac{0}{3.0924} \Rightarrow \beta_1 = 0$$

$$\beta_0 = 1.073 - (0)\mu_U \Rightarrow \beta_0 = 1.073$$

$$\sigma^2_{U,Y} = 0.5237 - (0)\sigma^2_U \Rightarrow \sigma^2_{U,Y} = 0.5237$$

At this point, the missing values in Y are replaced with the mean value $\hat{Y} = 1.073$. The aim of this step is to generate imputed values for the output Y in order to determine sufficient statistics which are: $\sum_{i=1}^{N} Y_i$, $\sum_{i=1}^{N} Y_i^2$, $\sum_{i=1}^{N} U_i \sum_{i=1}^{N} Y_i$ and Y_i^2, which is the squared output data:

$$Y_i^2 = \hat{Y}_i^2 + \sigma^2_{U,Y} = 1.073^2 + 0.5237 = 1.675$$

Table 3.1 shows the calculations for the first iteration of the expectation step. After each expectation step, a maximization step follows which makes use of the sufficient statistics generated from the E-step to develop new parameters for the linear model (see Table 3.1). It further substitutes the values in Table 3.2 using equations 3.14 and 3.18.

$$\mu_1 = \begin{bmatrix} \mu_U \\ \mu_Y \end{bmatrix} = \begin{bmatrix} 3.758 \\ 1.073 \end{bmatrix}$$

$$\Sigma_1 = \begin{bmatrix} \sigma^2_U & \sigma_{U,Y} \\ \sigma_{Y,U} & \sigma^2_Y \end{bmatrix} = \begin{bmatrix} 3.0924 & 1.07 \\ 1.07 & 0.4585 \end{bmatrix}$$

Table 3.1 Calculations of the first expectation step

U_i	U_i^2	Y_i	Y_i^2	U_iY_i
1.4	1.96	0.2	0.04	0.28
1.4	1.96	0.2	0.04	0.28
1.3	1.69	0.2	0.04	0.26
1.5	2.25	0.2	0.04	0.3
1.4	1.96	0.2	0.04	0.28
1.7	2.89	0.4	0.16	0.68
1.4	1.96	0.3	0.09	0.42
1.5	2.25	0.2	0.04	0.3
1.4	1.96	0.2	0.04	0.28
1.5	2.25	0.1	0.01	0.15
1.5	2.25	0.2	0.04	0.3
1.6	2.56	0.2	0.04	0.32
1.4	1.96	0.1	0.01	0.14
1.1	1.21	0.1	0.01	0.11
1.2	1.44	0.2	0.04	0.24
1.5	2.25	0.4	0.16	0.6
1.3	1.69	0.4	0.16	0.52
...	...	...	...	...
6.1	37.21	1.073	1.151329	6.5453
5.6	31.36	1.073	1.151329	6.0088
5.5	30.25	1.073	1.151329	5.9015
4.8	23.04	1.073	1.151329	5.1504
5.4	29.16	1.073	1.151329	5.7942
5.6	31.36	1.073	1.151329	6.0088
5.1	26.01	1.073	1.151329	5.4723
5.1	26.01	1.073	1.151329	5.4723
5.9	34.81	1.073	1.151329	6.3307
5.7	32.49	1.073	1.151329	6.1161
5.2	27.04	1.073	1.151329	5.5796
5	25	1.073	1.151329	5.365
5.2	27.04	1.073	1.151329	5.5796
5.4	29.16	1.073	1.151329	5.7942
5.1	26.01	1.073	1.151329	5.4723

As observed, the values that are imputed in the output parameter Y remains the same as the mean value. This is due to the fact that the mean value of the incomplete data is equal to the intersection parameter. However, there was a slight change in the variance of the output Y after the imputation, owing to the equations for computing the sufficient statics. The next

E-step will begin after the first iteration by using the new values of the mean vector and covariance matrix to generate a new linear model in the next maximization step.

The procedures involved in the first iteration of the E-step is repeated. The following results are produced by entering the new parameter values in the equation for sufficient statistics.

$$\beta_1 = \frac{1.07}{3.0924} \Rightarrow \beta_1 = 0.346$$

$$\beta_0 = 1.073 - (1.07)\mu_U \Rightarrow \beta_0 = 0.003$$

$$\sigma^2_{U,Y} = 0.4585 - (0.346)^2\sigma^2_U \Rightarrow \sigma^2_{U,Y} = 0.3388$$

Table 3.2 Sufficient statistics of the first iteration of the E-step

$\sum_{i=1}^{N} U_i$	$\sum_{i=1}^{N} U_i{}^2$	$\sum_{i=1}^{N} Y_i$	$\sum_{i=1}^{N} Y_i{}^2$	$\sum_{i=1}^{N} U_i Y_i$
563.8	2583	160.987	241.555	765.818

In the next step, the predicted values of ($\hat{Y}$) will be unequal to the mean value. This is because the parameter β_1 that will be substituted in equation 3.14 will have a non-zero value. The results of the second E-step can be see in Table 3.3. As usual, the maximization step will follow immediately after the expectation step. Table 3.4 shows the sufficient statistics that was yielded from the second iteration of the E-step. The M-step will repeat the process and predict new values for the mean and covariance vector as follows:

$$\mu_2 = \begin{bmatrix} \mu_U \\ \mu_Y \end{bmatrix} = \begin{bmatrix} 3.758 \\ 0.373 \end{bmatrix}$$

$$\Sigma_2 = \begin{bmatrix} \sigma^2_U & \sigma_{U,Y} \\ \sigma_{Y,U} & \sigma^2_Y \end{bmatrix} = \begin{bmatrix} 3.0924 & 0.92 \\ 0.92 & 0.5127 \end{bmatrix}$$

Table 3.3 Calculations of the second expectation step

U_i	U_i^2	Y_i	Y_i^2	U_iY_i
1.4	1.96	0.2	0.04	0.28
1.3	1.69	0.2	0.04	0.26
1.5	2.25	0.2	0.04	0.3
1.4	1.96	0.2	0.04	0.28
1.7	2.89	0.4	0.16	0.68
1.4	1.96	0.3	0.09	0.42
1.5	2.25	0.2	0.04	0.3
1.4	1.96	0.2	0.04	0.28
1.5	2.25	0.1	0.01	0.15
1.5	2.25	0.2	0.04	0.3
1.6	2.56	0.2	0.04	0.32
1.4	1.96	0.1	0.01	0.14
1.1	1.21	0.1	0.01	0.11
1.2	1.44	0.2	0.04	0.24
1.5	2.25	0.4	0.16	0.6
1.3	1.69	0.4	0.16	0.52
...	...	...	...	...
6.1	37.21	0.3732	0.13927824	2.27652
5.6	31.36	0.3732	0.13927824	2.08992
5.5	30.25	0.3732	0.13927824	2.0526
4.8	23.04	0.3732	0.13927824	1.79136
5.4	29.16	0.3732	0.13927824	2.01528
5.6	31.36	0.3732	0.13927824	2.08992
5.1	26.01	0.3732	0.13927824	1.90332
5.1	26.01	0.3732	0.13927824	1.90332
5.9	34.81	0.3732	0.13927824	2.20188
5.7	32.49	0.3732	0.13927824	2.12724
5.2	27.04	0.3732	0.13927824	1.94064
5	25	0.3732	0.13927824	1.866
5.2	27.04	0.3732	0.13927824	1.94064
5.4	29.16	0.3732	0.13927824	2.01528
5.1	26.01	0.3732	0.13927824	1.90332

Table 3.4 Sufficient statistics of the second iteration of the E-step

$\sum_{i=1}^{N} U_i$	$\sum_{i=1}^{N} U_i{}^2$	$\sum_{i=1}^{N} Y_i$	$\sum_{i=1}^{N} Y_i{}^2$	$\sum_{i=1}^{N} U_iY_i$
563.8	2583	147.691	222.326	693.459

When considering the fully observed data, the parameter values converged after the first iteration. This is because the model parameters were sufficient enough to enable the log-likelihood function to get to the height of the curve. Contrary to that, the mean vector and

covariance matrix of the incomplete output data Y, did not converge even after the second iteration. This is because the algorithm will require several iterations to reach the settling value in the presence of missing values. The number of iterations however depend on the amount of missing values present. The final outcome of the EM imputation process can be see in Figure 3.5.

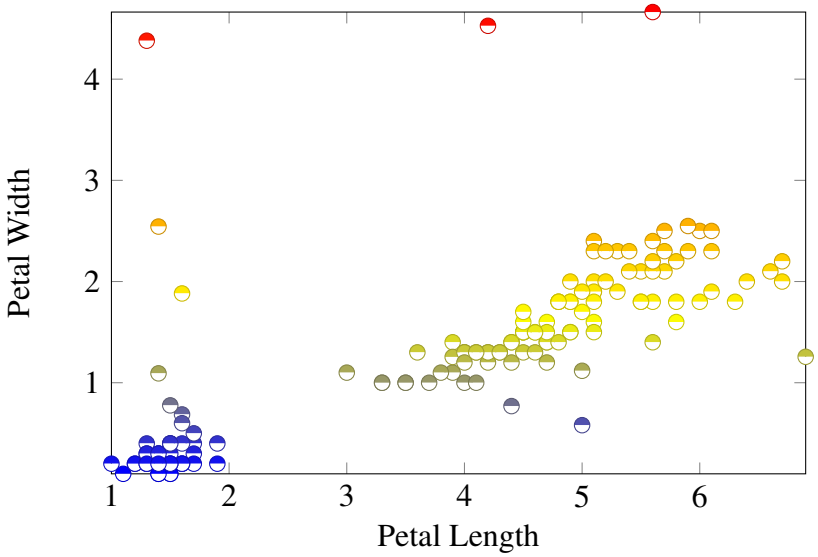

Fig. 3.5 Expectation Maximisation Imputation

3.3.3 Discussion

In this chapter of the thesis, the author identifies approaches to handling missing data. First of all, a highlight of traditional approaches to missing data treatment was presented and their shortcomings were identified. Modern approaches to missing data treatment were also discussed and a review of the merits of imputation, as a strategy for handling missing data was also reviewed. As missing data imputation is the focus of this research, the author conducted a review on various statistical and machine learning techniques for handling missing data and presented a detailed explanation of statistical missing data treatment methods.

To further demonstrate the implications of missing data treatment methods, missing data was simulated on the iris dataset obtained from the UCI machine learning repository and

imputation was performed using different techniques. It can be seen that identifying the correct approach to treating missing data is important as this will have an effect on the further analysis conducted on such data sets. Overall, from the statistical approaches considered, the regression imputation approach showed promising results by yielding a correlation value of 0.974 between the impute and response variable. Imputation using mean imputation also based imputed values around the central tendency of the distribution. This approach however fails to account for the variability in the distribution, which will result in biased estimates during analysis. The EM imputation technique however showed the least performance for the mechanism and pattern of missing data considered in this example.

Chapter 4

Optimal Missing Data Imputation for Downstream Regression in Air Quality Monitoring Stations

4.1 Introduction

The rapid increase in the use of IoT technology has resulted in a vast amount of data being collected, mostly from sensor devices. The IoT relies on data collected by various end devices, and data that is collected instantly by physical objects that incorporate sensors and network connectivity (Lee et al., (2021); Al-Aqrabi et al., (2020)). As a result, there has been a huge surge in the amount of data being generated and sent over the internet. When handling large amounts of data, it has become rather usual to come across large amounts of missing values in the data. Sensor data frequently contains missing values due to data collection and transmission errors (Agbo et al., 2022). Due to common-mode failures, data with missing values continues to emerge as a long-standing difficulty in the IoT architecture, potentially resulting in bias and loss of precision. These systems rely on data analytics applications that make decisions based on end device data.

Atmospheric pollutants in urban areas are considered as a main factor that has led to the increase in respiratory sickness among citizens. Some of these pollutants, e.g., benzene have previously induced cancers when citizens are exposed for prolonged periods of time (De Vito et al., 2008). Hence, it is important to accurately estimate the distribution of these pollutants as they are relevant for traffic management and assists in the design and mobilization of plans to tackle these problems.

Currently, air pollution monitoring in urban areas is essentially carried out using networks of fixed stations that are spatially distributed. These stations can accurately and selectively estimate the concentration of various atmospheric pollutants using Low Cost Sensor (LCS) devices. However, it is sometimes challenging to adequately deploy these networks due to their size and cost (Mazzeo and Venegas, (2005); Agbo et al., (2022)).

It has become possible to acquire spatio-temporal data variables for urban atmospheric pollutants by making use of LCS that are strategically placed in polluted areas. However, the data collected by these sensors are exposed to numerous issues such as drifts, bias and loss of data due to equipment failures (Loy-Benitez et al., 2020). The issue of missing data is a prevalent problem that affects most sensor network domains and other real-life datasets such as traffic (Chen et al., 2019) geo-informatics (Sanjar et al., 2020), and industrial applications (Ehrlinger et al., 2018).

If missing data is not handled correctly, it can have a significant impact on the veracity of data-based insights. By minimising the sample size and thereby introducing bias, it may limit the study's final results (Agbo et al., 2022).

4.2 Strategies for Model Assignment and Evaluation

In this section, the author draws inspiration from the works of Powell (2012). The vast amount of modelling methods and techniques identified in literature shows that various candidate models are available which may show a good representation of the data **p** but still

show some level of differences. For instance, some candidate models show differences in their covariate risk factors or the way they are imputed into the model. In a setting where the implementation considers a Bayesian analysis, the models may vary in consideration to the chosen probable distributions that are antecedent. There is also a possibility that a model can differ through the probability distribution identified in **p** or through $g(.)$, which is the link function. These potential variations identified in models may lead to changes in the results generated from the analysis. A number of tools may be useful for processes involved in model selection and assessment. A brief description of these methods are highlighted in this chapter. It is however noteworthy that despite the usefulness of such methods, it is equally important to make use of personal judgment and experience in the selection and assessment processes.

In some studies, the overall goal of estimating model parameters may not be for the purpose of drawing conclusions. On the other hand, models may be built in some cases for the purpose of making accurate predictions. For instance, some models may be built to forecast future events like shares, stock prices etc., while some are developed to make predictions about spatial data. In studies focusing on air quality data, models are built to make spatial predictions which will be covered in this chapter.

4.2.1 Candidate Model Selection using Probabilistic Measures

Tools used for model selection assist in choosing between different candidate models. These models vary only by their covariates. Therefore, in their simplest form, such models are said to be nested. On the other hand, the differentiation between models may also be affected by more complex entities.

Before we describe methods that are used for comparing models, we first of all define a concept proposed by Nelder and Wedderburn (1972), called deviance. The deviance, which is also referred to as the log-likelihood (ratio) statistics, is a representation of the difference

between a saturated model and a candidate model. A saturated model shows the same link function and distribution as the candidate. However, a saturated model possesses the highest amount of covariates. Therefore, this type of model allocates all the variations present in **p** to the portion of the model that is fitted. The deviance is therefore represented by

$$Dev(\mathbf{p}) = -2[log(f(p|\hat{\theta})) - log(f(p|\hat{\theta}_s))], \tag{4.1}$$

where $\hat{\theta}$ represents the parameter values fitted in the candidate model and $\hat{\theta}_s$ represents the parameter values fitted in the saturated model. Candidate models can therefore be compared by calculating their individual deviances. A candidate model showing the smallest deviance is voted as a better fit for the data in question.

The Akaike's Information Criterion (AIC) (Akaike, 1974) is another important model selection criteria. This model shows some similarities to deviance but differs in some ways as it penalises models with excessive parameters using a penalty term. The AIC model selection criteria is given by;

$$AIC = 2_m - 2log(f(\mathbf{p}|\hat{\theta})). \tag{4.2}$$

A similar selection criteria which also follows the likelihood function is Bayesian Information Criterion (BIC) (Schwarz, 1978). This can be given by

$$BIC = mlog(n) - 2log(f(\mathbf{p}|\hat{\theta})), \tag{4.3}$$

where n represents the number of datapoints.

4.2.2 Model Evaluation

The ability of a model to clearly describe the variation in a given data **p** depends on the level of the variations it allocates to the model that is fitted and the variation assigned to residual

components. This is referred to as the unexplained variation. For a model to show good performance, it should therefore reflect a small residual component as compared to other models. A brief description of some techniques for model evaluation can be seen in the following subsections.

Assessing the Adequacy of a Fitted Model

To measure the adequacy of a fitted model, a Pearson's chi-squared test (often called goodness-of-fit measure) can be used. This test is a measure of the distance between the value of the fitted model and **p**. This can be represented by

$$T = \sum_{t=1}^{n} \frac{(p_t - \hat{\mu})^2}{Var(\hat{\mu}_t)} \sim V^2_{n-p} \tag{4.4}$$

For a model to be adequate, the test statistics will have an approximate distribution of V^2_{n-p}, where n signifies the number of observations and p represents the effective number of parameters.

Checking the Posterior Predictive Model

This method is a Bayesian tool for checking model accuracy, presented by (Rubin, 1984). If the model fits well with the data concerned, the data replicated from the model will show some similarities with the observed data. In order to measure the fit of the model, the observed data is compared with samples drawn from the posterior predictive of the replicated data. Therefore, the posterior predictive distribution is given by

$$f(\mathbf{p}^{rep}|\mathbf{p}) = \int f(\mathbf{p}^{rep}|\theta, \mathbf{p}) d\theta \tag{4.5}$$

where $\mathbf{p}^{rep}$ represents the replicated data which could have been observed. Simulation can be used to evaluate the posterior distribution, drawing a sample of θ from its given

posterior distribution and sampling $\mathbf{p}^{rep}$ from $f(\mathbf{p}|\theta)$ having prescribed the values of θ that are sampled. A scalar summary of the data and parameters, indicated by a test statistic $T(p,\theta)$, can be used to measure any discrepancies that occur between the data and the model. Any deficiency in fitting the data in the posterior distribution can be calculated using the posterior predictive p-value given by

$$p\text{-value} = P(T(p^{rep},\theta) \geq T(p,\theta|p)), \tag{4.6}$$

which evaluates the probability of the test statistic obtained from the data recreated as compared to the observed data.

4.2.3 Model Predictive Capability

Mostly, statistical models and techniques are used for making predictions. Therefore, various tools exist for the purpose of assessing the predictive capabilities of various models, such as cross-validation, median absolute deviation and prediction bias. These methods aid in evaluating model performance such as the methods described previously and would be used to describe the performance of the proposed model later in this chapter.

Cross-validation is a technique used for evaluating model performance. The idea of this technique involves splitting the data into two distinct sets (training and testing/validation set). When a model is fitted on the training set, the test set is predicted using the resulting parameter estimates from the training data. This concept is often referred to as Predicted Residual Sum of Squares (PRESS) (Wand, (2000); Powell, (2012)) given by

$$CV = \sum_{t=1}^{n}\sum_{j=1}^{q}(p_{t,j} - \hat{p}_{t,j})^2, \tag{4.7}$$

where $p_{t,j}$ represents the real observations recorded at time $t = 1,\ldots,n$ and at location $j = 1,\ldots,q$ and $\hat{p}_{t,j}$ indicates observations that are predicted for the same period of time and

locations, obtained by the model that excludes $p_{t,j}$. Some specific segments of the data could also be applied for validation. For example, a strategy called leave-one out cross-validation can exclude a sole observation in a dataset before the model is fitted. The process then iterates for each record observed in the dataset.

The prediction bias assesses the total bias in the model predictions and is represented as;

$$\text{PB} = \text{Median}_{t,j}\{\hat{p}_{t,j}^{-j} - p_{t,j}\}. \tag{4.8}$$

Another similar method called median absolute deviation can be used to assess model prediction in place of root mean squared error (RMSE). This can be represented by

$$MAD = \text{Median}_{t,j}\{|\hat{p}_{t,j}^{-j} - p_{t,j}|\}, \tag{4.9}$$

this measures the mean error that exists between the predicted and observed data.

4.3 Methodology

4.3.1 Formulation of the Problem

For the missing data problem in this case, the author considers a data set where the response variable is generated and observed for each time point through environmental sensor devices. Due to various reasons such as sensor failures, loss of communication, etc., covariate values may be partially or completely missing for instances at different time points. Lets take $P = \{P_{ijk}\}_{i=1}^{n}$ as an $n \times t \times p$ matrix where P_{ijk} is the k-th covariate of the i-th instance at a time point j. This thesis assumes a matrix with n instances, p covariates and t time points. Likewise, the author also considers $V = \{V_{ij}\}_{i=1}^{n}$ as the matrix of $n \times t$ responses where the i-th response is represented by V_{ij} for a time point j. V represents continuous variables that are observed over t timepoints. There are n observations for t timepoints. The author

considers the sensor data $\{(P_{ijk}, V_{ij})\}_{i=1}^{n}$ with n instances. The data is further satisfied based on the following model:

$$E(V_{ij}|P_{ij}) = f(P_{ij}), i = \{1,2,\ldots,n\}, j = \{1,2,\ldots,t\}, \tag{4.10}$$

where $f(.)$ represents a real-valued continuous function, $P_{ij} = \{P_{ijk}\}_{k=1}^{m}$ has some entries missing and the values in V are completely observed.

Missing entries for the data matrix P is denoted using the indicator matrix $D_{ij} = \{D_{ijk}\}_{i=1}^{n}$ which has the same dimensionality as P and is represented as:

$$D_{ijk} = \begin{cases} 0, & P_{ijk} \text{ is missing} \\ 1, & \text{otherwise} \end{cases} \tag{4.11}$$

Lets assume the matrix contains non-mixed-type data. Without any loss of generality in the matrix, lets also assume that each pair (i, j), $P_{ij} = \{P_{ijk}\}_{k=1}^{m}$ contains m continuous features for $j \in \{m_0 + 1, \ldots, m_0 + m_1\}$ such that $m_0 + m_1 = m$. Let the j-th continuous variables constituting the $(m_0 + j)$-th feature of P, indexed by $j \in \{1, \ldots, m_1\}$ take values from a continuous set $C_j \subset \mathbb{R}$.

Lets assume that the data follows a missing at random (MAR) mechanism, where the distribution of D does not rely on the missing values in P but depends on the observed values as seen below;

$$M(D|P) = M(D|P_{obs}), \text{ for all } P_{miss} \tag{4.12}$$

This thesis proposes a clustering-based approach for imputing the $n \times t \times p$ matrix P, since it will show complex interactions that exist between the m dimensional covariates.

4.3.2 Correlation and Covariance Matrix for Multivariate Data

The covariance matrix for the variables X_{it} and X_{j-p} of the data series X_t, does not depend on the timestamp t. It is however a function of the lag p, where:

$$Cov(X_{it}, X_{j-p}) = E[(X_{it} - M_i)(X_{j,t-p} - M_j)^T] = \gamma_{ij}(p) \tag{4.13}$$

the n $\times$ p covariance matrix can be expressed as follows:

$$\gamma(p) = E[(X_{it} - M_i)(X_{j,t-p} - M_j)^T] = \begin{bmatrix} \Gamma 11(p) & \Gamma 12(p) & \cdots & \Gamma 1n(p) \\ \Gamma 21(p) & \Gamma 22(p) & \cdots & \Gamma 2n(p) \\ \vdots & \vdots & \vdots & \vdots \\ \Gamma n1(p) & \Gamma n2(p) & \cdots & \Gamma nn(p) \end{bmatrix} \tag{4.14}$$

furthermore, the cross-correlation matrix for the n $\times$ p matrix is expressed as:

$$\delta(p) = U^{-1/2}\gamma(p)U^{-1/2} = \begin{bmatrix} \delta 11(p) & \delta 12(p) & \cdots & \delta 1n(p) \\ \delta 21(p) & \delta 22(p) & \cdots & \delta 2n(p) \\ \vdots & \vdots & \vdots & \vdots \\ \delta n1(p) & \delta n2(p) & \cdots & \delta nn(p) \end{bmatrix} \tag{4.15}$$

For p = 0,1,2, …, the square root of the diagonal covariance matrix represents the U vector, where:

$$U^{-1/2} = Diag\{\frac{1}{\sqrt{\Gamma 11(0)}}, \ldots, \frac{1}{\sqrt{\Gamma 11(0)}}\} \tag{4.16}$$

$$\delta_{ij}(p) = Corr(X_{it}, X_{j,t-p}) = \Gamma_{ij}(p)/\sqrt{[\Gamma_{ii}(0)\Gamma_{jj}(0)]} \tag{4.17}$$

where $\Gamma_{ii} = Var(X_{it})$. Therefore, $i = j$ and $\delta_{ii}(p) = \delta_{ii}(-p)$ represents the auto-correlation of the i^{th} series X_{it} and $i \neq j, \delta_{ii}(p) = \delta_{ii}(-p)$ represents the cross-correlation between X_{it}

and X_{jt}. It is noteworthy that $\Gamma_{ij}(p) = \Gamma_{ji}(-p)$. Therefore,

$$\left\{ \begin{array}{l} \gamma(p)^T = \gamma(-p) \\ \delta(p)^T = \delta(-p), \end{array} \right\} \tag{4.18}$$

Also, the cross-correlation and cross-covariance matrices $\delta(p)$ and $\gamma(p)$ respectively, posses the properties of positive definiteness, where

$$\sum_{i=1}^{n} \sum_{j=1}^{n} m_i^T \gamma(i-j) m_j \geq 0 \tag{4.19}$$

and

$$\sum_{i=1}^{n} \sum_{j=1}^{n} m_i^T \delta(i-j) m_j \geq 0 \tag{4.20}$$

Considering all non-negative values k and the n-dimensional vectors $m_1, \ldots, m_k$, which follows because $Var(\sum_{j=1}^{n} m_i^T X_{t-i}) \geq 0$

4.4 Proposed $k-$ Best Fit Missing Value Imputation Model

An imputation model based on k-Means algorithm is proposed, which is capable of choosing the most optimal imputation method from a selection of predefined sub-techniques. Three stages are considered in the imputation model; firstly, the algorithm partitions the incomplete data into different groups based on the k-means algorithm. Secondly, missing values in each independent cluster is estimated based on the observed values within each group. In the third stage, the algorithm selects the most suitable imputation technique for each group based on predefined imputation techniques fitted in the model. Highly correlated features were considered in the imputation model based on Equation 4.15.

In the next sections, the author presents the stages involved in the proposed imputation approach. It is noteworthy that the proposed technique assumes a univariate data series **P** which shows a sequence of real numbers **P** = $p_1, p_2, \ldots, p_n$ where N represents the length

of the series. In addition, the algorithm takes a missing sequence $V_{i,l} \in \mathbf{P}$ which is a set of continuous missing data NA where the length l ranges from i to i + l.

4.4.1 Partitioning the Dataset

According to Zhang et al., (2015), more accurate imputation estimates could be derived when similar records are used to estimate missing instances. However, Zhao et al., (2018) argued that current clustering algorithms do not perform optimally in the presence of missing data as missing values constitute major uncertainties in a dataset, therefore affecting the usability and accuracy of existing clustering algorithms. Therefore, before grouping the data, missing values were first of all pre-imputed using distinct values, after which the dataset was split into k = 3 distinct groups using Algorithm 1.

Algorithm 1 k-Means Algorithm

Input: Incomplete matrix X (n x p). pre-imputed with distinct values
Output: Cluster labels C_{ik} with points $\{p_1, \ldots, p_n\}$

Pre-impute missing values in matrix X (n x p) with distinct single imputation values
Select initial centers k at random $C = \{c_1,....,c_k\}$
while convergence criteria is not met **do**
    assignment step:
    for i = 1,. . . ,N **do**
        locate nearest centroid $c_k \in C$ to points $\{p_i, \ldots, p_n\}$
        append points $\{p_i, \ldots, p_n\}$ to the set C_k centroid
        Update:
        for i = 1,. . . ,k **do**
            $c_i \rightarrow$ center of all points in C_i
        end for
    end for
end while
Assign cluster label (C_{ik}) to points $\{p_1, \ldots, p_n\} \in$ X

4.4.2 Defining the Imputation Strategy

The second stage is initiated after the dataset has been grouped into clusters with similar records. A random forest model is trained on each group before aggregating the data. This ensured that strong predictors $P_s = 1,\ldots,m$ were used in the training process.

In the proposed approach, lets assume an $n \times p$-dimensional matrix where $P = (P_1,\ldots,P_n)$. The rf algorithm is first of all used to fill in missing observations in each partition created by the k-Means algorithm. A built-in routine is added to the rf algorithm for handling missing values by considering the frequency of values in the recorded variables with their rf proximities after initially training the model on the dataset pre-imputed with mean values (Breiman, 2001). This approach mostly requires a fully observed response variable before the rf model can be trained. However, the proposed approach directly estimates the missing values by using the rf model on a training set containing only the observed data, with P, being the matrix with complete data and P_s representing the sample with missing vales $i_{miss}^{(s)} \subseteq \{1,\ldots,n\}$. To better understand the training process, the data is separated into four parts as described below:

1. $v_{obs}^{(s)} \rightarrow$ representing the values that are present in the variable P_s
2. $v_{miss}^{(s)} \rightarrow$ representing the values that are missing in the variable P_s
3. $p_{obs}^{(s)} \rightarrow$ representing the observations, $i_{obs}^{(s)}$ = {1,. . . ,n }\$i_{miss}^{(s)}$ of the predictor variable in P_s
4. $p_{miss}^{(s)} \rightarrow$ representing the observations, $i_{miss}^{(s)}$ of the predictor variable in P_s

It is important to note that $i_{obs}^{(s)}$ points only to the observed values in P_s. Therefore, $p_{obs}^{(s)}$ is not completely observed and likewise, $p_{miss}^{(s)}$ is not completely missing.

Similar to the work in Stekhoven and Bühlmann (2012), the process is initiated by pre-imputing the missing values in X with the mean of the distribution, after which the

predictors $X_s = 1,\ldots,p$ are stacked in ascending order considering the amount of values that are missing. Each missing value in X_s is then imputed by first of all fitting the rf on the response $y_{obs}^{(s)}$ and predictor variable $x_{obs}^{(s)}$. Next, the trained rf model is applied to $x_{miss}^{(s)}$ to predict the missing values of $y_{miss}^{(s)}$. The imputation process is repeated until the set stopping criterion (γ) has been met. This is achieved when the difference between the most recent imputed data matrix and the old matrix has an increase for the first time, considering the variable types present. Lets take the $n \times p$ matrix to be a set of continuous variables in the proposed approach. Therefore, the difference in the new and previous imputed matrix $\boldsymbol{N}$ is defined as:

$$\Delta_N = \frac{\sum_{j\in N}(X_{new}^{imp} - X_{old}^{imp})^2}{\sum_{j\in N}(X_{new}^{imp})^2} \tag{4.21}$$

This step is followed by two additional imputation phases where missing values in each cluster is also estimated using a kNN and linear regression model.

Still considering an $n \times p$ matrix P_s, the procedure for next imputation phase is described as follows;

Algorithm 2 k-NN Imputation Strategy

Input: Cluster labels C_{ik} with points $\{p_1,\ldots,p_n\}$
Output: Dataset with Imputed values X_s.

1: The missing values in each cluster matrix C_i are located.
2: The kNN vectors are defined by; $x_{(1)}^D,\ldots,x_n^D$ with $d(x_i,x_{(1)}) \leq,\ldots,\leq d(x_i,x_{(k)})$, where $x_{(1)}^D$ represents the rows of the matrix X^D, and $d(x_i,x_{(k)})$ is the distance given by equation 3.19.
3: For each point (y) in C_i, the distance (x,y) between the missing point and the nearest imputed value is stored in a similarity array (S).
4: The array (S) is sorted in descending order and the top K data for (y) in C_i is selected for imputation.

The linear regression imputation process follows, as described below;

1. For each matrix C_i the data was split into four parts similar to the rf method where a regression model was trained on the response $y_{obs}^{(s)}$ and predictor variable $x_{obs}^{(s)}$ which solves the problem in 3.5.

2. The trained regression model is then used to predict the missing values in X_s.

4.4.3 Selecting the Best Fit Estimation

After computing the missing values, the next stage is the selection of the most suitable imputation method within each group. For each data matrix C_i, the selection is done by estimating the error between the previous imputation $y_{pre,i}$ and current imputation $y_{cur,i}$ based on the equation below;

$$err = \sqrt{\frac{1}{n}\sum_{i=1}^{n} y_{pre,i} - y_{cur,i}} \tag{4.22}$$

The result of the rf imputation is set as the threshold γ and $y_{pre,i} \equiv \gamma$ is placed as the initial value of the previous estimate as described in Algorithm 3.

Lastly, a reverse error score function RES(r) is used to obtain the final imputation sequence. This is based on two error calculations between the previous imputation estimate with the lowest error and the given threshold γ. A sequence that gives the lowest error score is chosen as the optimal imputation estimate for the given distribution.

Lets assume a reverse error score function RES(r) representing the error between γ and the final sequence β_{Ci} in each group C_i. The equation can be derived by:

$$M_\gamma = \frac{\partial(\gamma)}{\partial_n} = \sqrt{\frac{\sum_{i=1}^{N}(X_\gamma - \hat{y}_{\beta_{ci}})^2}{n}} \tag{4.23}$$

$$M_{\beta_{ci}} = \frac{\partial(\beta_{ci})}{\partial_n} = \sqrt{\frac{\sum_{i=1}^{N}(\hat{y}_{\beta_{ci}} - X_\gamma)^2}{n}} \tag{4.24}$$

$$RES(r) = M(\gamma, \beta_{ci}) = \left[\frac{\partial(\gamma)}{\partial_n} \; \frac{\partial(\beta_{ci})}{\partial_n} \right] \tag{4.25}$$

where γ is the imputation threshold for C_i and β_{Ci} is the best estimate from the previously chosen imputation techniques.

Algorithm 3 k-Best Fit Missing Value Imputation ($k-BFMVI$)

Input: Cluster labels C_{ik} with points $\{p_1, \ldots, p_n\}$
Output: Dataset with Imputed values X_s.

while 1 **do**
  for v $\in C_i$ **do**
    RF_i^1 = missForest ($p_i \in C_i$)
    RI_i^1 = regression ($p_i \in C_i$)
    kNN_i^1 = k-Nearest Neighbour ($p_i \in C_i$)
  end for
  $err_{ci} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} \gamma_{pre,i} - \alpha_{cur,i}}$
  if $RI_{err} \leq kNN_{err}$ **then**
    $RI_i^1 \rightarrow \beta_{Ci}$
  else$kNN_i^1 \rightarrow \beta_{Ci}$ // Store imputed value in a temporary array
  end if
  Get the imputation sequence ($M^{'}$) and compute a reverse error score function RES(r):
  for $v = 1 : k$ **do**
    Get the reverse error score RES(r) using Eq.4.25
    if $\beta_{Ci,err} \leq \gamma_{err}$ **then**
      $\beta_{Ci} \mapsto C_i$
    else$\gamma \mapsto C_i$
    end if
  end for
end while
Return imputed dataset X_s;

4.5 Simulation Studies

In this section, the author assesses the performance of the proposed k-Best Fit Missing Value Imputation (k-BFMVI) algorithm for recovering missing instances and improving the prediction accuracy of the data, and a report of the computational complexity of the

algorithm is presented. This thesis compares the proposed techniques with 6 well established imputation methods as follows:

- **Expectation Maximization Imputation (imp.em)**: This technique was proposed by Dempster et al., (1977) to handle the missing data problem. The general idea behind the EM algorithm is to iteratively estimate parameters required to predict the missing values in a given distribution by calculating the mean and variance between parameters (Bashir, 2019). The application of the EM algorithm usually aims at solving missing data problems. However, research has also shown the use of this algorithm in solving complex problems for complete data sets. For example, multilevel linear models, structural equation model and finite mixture (Liang and Bentler, (2004); Muthén and Shedden, (1999); Neale et al., (1999);Raudenbush and Bryk, 2002).

  The EM algorithm constitutes iterative processes built on two steps: the expectation step ("E-step") and maximization step ("M-step"). The initial values available in a system are required to initiate the estimation process. The E-step begins with the construction of a linear regression model, using the co-variance matrix and initial mean vector to produce estimates for missing values based on the values from the observed data. The M-step follows after the E-step and produces new parameter values for the data that has been estimated. The algorithm saves the last co-variance matrix and mean vector to determine the next E-step and builds a new regression model from the results which is then used to determine new estimates for missing values. The M-step will subsequently run again to determine new parameters from the updated estimates. The EM algorithm will iterate these two steps until the values of the co-variance matrix and mean vector no longer changes or converges, where the converged value corresponds to that of the value of the maximum likelihood estimates. The number of iterations in the EM algorithm is dependent on the number of missing values in the dataset (Bilmes,

1998). Research conducted by Dempster et al. (1977) shows a detailed description of the EMI algorithm.

- **K-Nearest Neighbour Imputation (imp.*k*nn)**: The KNN algorithm is also employed to benchmark the proposed technique. The KNN method is a machine learning algorithm which approaches imputation by classifying the k closest observed values to missing values and uses the average of these k nearest neighbours to impute missing values going by the distance measure between points (Maillo et al., 2017). Various distance factors have been applied to the KNN algorithm in research but for the purpose of this study, the Euclidean distance factor is used, which is the most widely used measure that maximizes efficiency and productivity of the algorithm (Amirteimoori and Kordrostami, 2010).

- **MissForest (imp.missforest)**: MissForest is another iterative technique based on the Random Forest (rf) algorithm. Previous research has shown the efficiency of this algorithm in handling multivariate missing values in high dimensional datasets in a computationally efficient manner. To impute missing values, the algorithm first trained the rf on the observed data using an iterative imputation scheme, after which the missing values were imputed iteratively.

- **Regression Imputation (imp.mice)**: The proposed technique was also benchmarked using imputation by multiple linear regression models based on multiple imputation by chained equations (MICE). Generally, a regression model is generated using the variables that presented high correlation results with the imputation variable and these variables were used to predict and replace the missing values on the target variable.

- **Mean Imputation (imp.mean)**: This is a straight forward method where missing value is imputed using the mean of the target distribution as described in section 3.2.2.

- **Stochastic Regression Imputation (stc.reg)**: Missing data imputation was also conducted using stochastic regressors. This technique is inspired by the assumption that a linear statistical model may not be tenable for stochastic processes. This strategy was implemented using Python's *linearmodel* library.

4.5.1 Dataset Description

For the purpose of conducting experiments, this thesis considered a dataset presented by De Vito et al. (2008), which is publicly available on the University of California Irvine (UCI) repository UCI. The dataset contains the concentration measures of target pollutants collected from a measurement site. These concentration values were used as a benchmark to tune a regression system that was designed to calibrate the response of the multi-sensor device. This device was configured to accommodate five metal oxide sensors and two solid state sensors to capture data on the temperature and relative humidity in the environment. The specified station provided concentration estimation values for CO (mg/m^3), C_6H_6 ($\mu g/m^3$), non-metalic hydrocarbons (NMHC) ($\mu g/m^3$), NO_2 ($\mu g/m^3$), NO_x (ppb). The data was sampled, showing hourly averages of the concentration results. However, the NMHC analyser went offline after 8 days, causing a series of missing data. Hourly average values of the multi-sensor device was sampled, showing concentration levels indicated by NO_x, CO, O_3, and NO_2 metal oxide (MOX) chemiresistors in addition to relative humidity and temperature sensors. More information on the MOX chemiresistor is presented in a research by (De Vito et al., 2008).

The original dataset contains real missing values on all columns ranging from as low as 3.91% to a high of 91%, with the highest rate recorded from the GT sensor which is disregarded in the analysis conducted in this thesis. A full description of the data set can be seen in Table 4.1.

Table 4.1 Air Quality Data Summary

	CO(GT)	PT08.S1(CO)	C6H6(GT)	PT08.S2(NMHC)	NOx(GT)	NO2(GT)	PT08.S4(NO2)	PT08.S5(NO3)
mean	2.182	1119.913	10.554	958.543	250.671	113.874	1452.648	1057.756
std	1.441	218.733	7.465	264.055	208.611	47.475	353.301	406.509
min	0.100	647.000	0.200	390.000	2.000	2.00	551.000	221.000
25 %	1.100	956.000	4.900	760.000	103.000	79.000	1207.000	760.000
50 %	1.900	1085.000	8.800	931.000	186.000	110.000	1457.000	1006.000
75 %	2.900	1254.000	14.600	1135.000	335.000	142.000	1683.000	1322.000
max	11.900	2040.000	63.700	2214.000	1479.000	333.000	2775.000	2523.000

4.5.2 Simulation Scenarios

Missing Completely at Random (MCAR)

Here, missingness is introduced in the dataset completely at random, based on the missing matrix D. The RMSE, and MAE scores for the missing rates ranging from 10% - 40% were recorded and a comparison of benchmark techniques and the $k-BFMVI$ method was also conducted. Table 4.2 shows the RMSE score of each algorithm at different missing data rates. It is observed from the results that the k-BFMVI algorithm generally shows the best imputation accuracy for the MCAR condition followed by the *imp.mice* imputation technique which is based on multiple linear regression. The MAE score in table 4.3 shows a close relationship between the outcome of the *imp.mice* and *imp.missforest* techniques. The *imp.em* algorithm however showed a poor performance with the highest RMSE and MAE scores of 5.933 and 2.734 respectively, when the missing data rate was at 40%. It is difficult to conclude on a generally acceptable threshold when comparing the error scores of imputation techniques as higher rates of missing values may still affect the overall scores of good performing imputation techniques. However, cross-validating imputation techniques on independent samples as seen in Figures 4.1 to 4.3, are effective for choosing a unique threshold that may be applicable to a unique case.

Table 4.2 RMSE Upon Imputation for Air Quality MCAR Dataset

Missing Rate	imp.*k*nn	imp.mice	imp.em	imp.missforest	*k*-BFMVI	stc.reg	imp.mean
10%	0.941595	0.802989	3.083324	0.835237	**0.011758**	1.261684	2.361304
20%	1.410407	1.206893	4.507154	1.231309	**0.029012**	2.076452	3.396705
30%	1.831899	1.630087	5.445520	1.631028	**0.215160**	2.767883	4.170062
40%	1.930442	1.722670	6.398958	1.746202	**0.169418**	3.529798	4.730075

Table 4.3 MAE Upon Imputation for Air Quality MCAR Dataset

Missing Rate	imp.*k*nn	imp.mice	imp.em	imp.missforest	*k*-BFMVI	stc.reg	imp.mean
10%	0.216405	0.193356	0.719108	0.197484	**0.000599**	0.311829	0.596996
20%	0.452831	0.392866	1.435791	0.392628	**0.001917**	0.709476	1.164325
30%	0.694035	0.61114	2.189204	0.60595	**0.006739**	1.156649	1.757839
40%	0.861461	0.771931	2.914241	0.77397	**0.006136**	1.736871	2.313817

Missing at Random (MAR)

In this section, the performance of the imputation techniques under the MAR condition is evaluated. The sensor data was simulated based on similar parameters taken from the MCAR condition and artificial missing data was simulated at random (MAR) by allowing the probability of missingness to depend on the observed values. In real-life, we sometimes come across both time-dependent and time-independent covariates which may not be missing. Here, lets assume X_{ij1}, X_{ij2} and X_{ij3} to be non-missing and the missingness of X_{ij4}, X_{ij5} and X_{ij6} to depend on the non-missing values of the covariate. The results in Table 4.4 and 4.5 shows that the $K-BFMVI$ algorithm performs better when data is MAR compared to benchmark techniques. Comparatively, the RMSE and MAE scores show close performance between the *imp.mice*, $imp.missforest$ and *imp.knn* techniques. Overall, among the comparative techniques, the *imp.knn* closely follows the performance of the proposed $k-BFMVI$ technique.

Table 4.4 RMSE Upon Imputation for Air Quality MAR Dataset

Missing Rate	imp.*k*nn	imp.mice	imp.em	imp.missforest	*k*-BFMVI	stc.reg	imp.mean
10%	0.763833	0.73903	2.793925	0.72689	**0.004776**	1.258882	2.222221
20%	1.156408	1.320681	3.727606	1.314526	**0.014769**	1.977577	3.118087
30%	1.680419	1.876043	4.709317	1.877224	**0.045882**	2.727725	3.905123
40%	2.140085	2.489410	5.933451	2.478726	**0.217910**	3.754916	4.831988

Table 4.5 MAE Upon Imputation for Air Quality MAR Dataset

Missing Rate	imp.*k*nn	imp.mice	imp.em	imp.missforest	*k*-BFMVI	stc.reg	imp.mean
10%	0.165567	0.193814	0.658888	0.184392	**0.000553**	0.314437	0.59275
20%	0.335533	0.469265	1.226727	0.455698	**0.001606**	0.710771	1.198543
30%	0.582721	0.806026	1.912380	0.793586	**0.003871**	1.203348	1.830785
40%	0.833084	1.197000	2.734274	1.178809	**0.012051**	1.848042	2.488014

Computational Complexity

Next, the computational complexity of imputation methods are compared, showing the time required to complete a cycle of imputation for the dataset with $n = 6941$ observations and $p = 8$ features across each identified missingness pattern. Simulations on each method were conducted on a single thread of a machine having an Intel Core 2 Duo (3.06 GHz) processor which is limited to 8 GB RAM. The results can be seen in Table 4.6 below.

The stochastic regression imputation technique was nearly instantaneous and therefore is not presented in the table. The imputation techniques finish quite quickly for both missing patterns when run on 30% missing rate. Mean imputation is a straight forward technique and therefore presents the best results in terms of complexity, followed by the *imp.mice* algorithm. Despite the imputation accuracy of the proposed method however, a trade off can however be seen between the complexity of the algorithm and the plausibility estimates generated by this technique as it shows a higher complexity. This is due to the overall computational

complexity of each sub-technique embedded in the algorithm. The **imp.em** shows an overall poor performance with a high imputation complexity in conjunction with a low accuracy.

Table 4.6 Average computational time for benchmark and $K-BFMVI$ imputation techniques at 30% Missing Rate

	Time (s)					
Missing Pattern	imp.*k*nn	imp.mice	imp.em	imp.missforest	$k-BFMVI$	imp.mean
MCAR	0.82	0.85	2.28	1.71	3.39	0.76
MAR	2.76	0.68	4.34	2.19	5.33	0.73

4.5.3 Performance on Downstream Regression Tasks

Next, the author assessed the performance of machine learning algorithms for regression that is trained on the data derived from each imputation method. The challenge of regression tasks on fully observed data sets also vary largely. In Figures 4.1 to 4.6, we can see the effect of the chosen imputation methods on the performance of regression algorithms for MCAR and MAR missing data scenarios. The imputation models were trained on supervised machine learning techniques (Multiple Linear Regression (MLR), Decision Tree (DT) and Random Forest (RF)). To evaluate the effect of the imputation techniques on the identified missing mechanisms, the analysis was conducted using the results of the imputed data at 30% missing rate.

It is observed that when trained on downstream regression algorithms, the $k-BFMVI$ technique shows the best performance against cross-validated benchmark techniques with significant improvements across the learning algorithms. For MCAR scenarios, it is seen that training the MLR model on $k-BFMVI$ presents the overall best performance with an RMSE and MAE scores of 0.031015 and 0.023258 respectively. Moreover, $imp.missforest$, $imp.mice$ and $imp.knn$ also recorded some improvements across the learning algorithms with R^2 score > 0.9 for all learning tasks. Comparatively, the $imp.em$ data trained on the learning

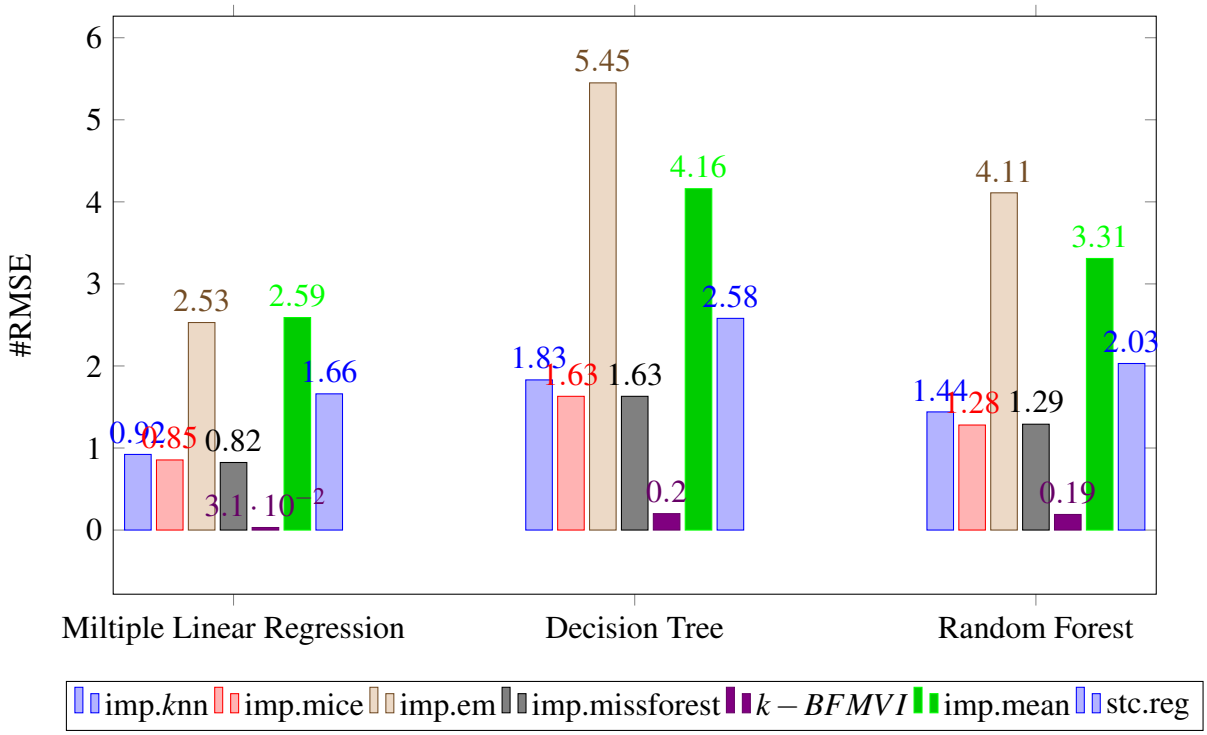

Fig. 4.1 RMSE Indicator for Supervised Learning Models Upon Imputation on MCAR Data

algorithms showed the weakest performance with an average score of 4.0326 and a lowest $R^2 < 0.5$ when trained on the DT algorithm.

Similarly, the performance of imputed data trained on the learning algorithms for MAR scenarios is also evaluated. The improvement of DT over MLR and RF can be observed in this scenario when trained on $k-BFMVI$ imputed data with a score of 0.040746. Comparatively, the *imp.em* imputed data trained on the learning algorithms still shows the weakest performance with the lowest score recorded from the DT learning algorithm.

4.5.4 Discussion

One of the main contributions of this chapter is the development of a robust imputation technique that is capable of making a choice among other sub-techniques embedded in the algorithm. This algorithm accommodates predictive models which describe conditional relationships that exist within a given distribution. By design, the proposed algorithm builds

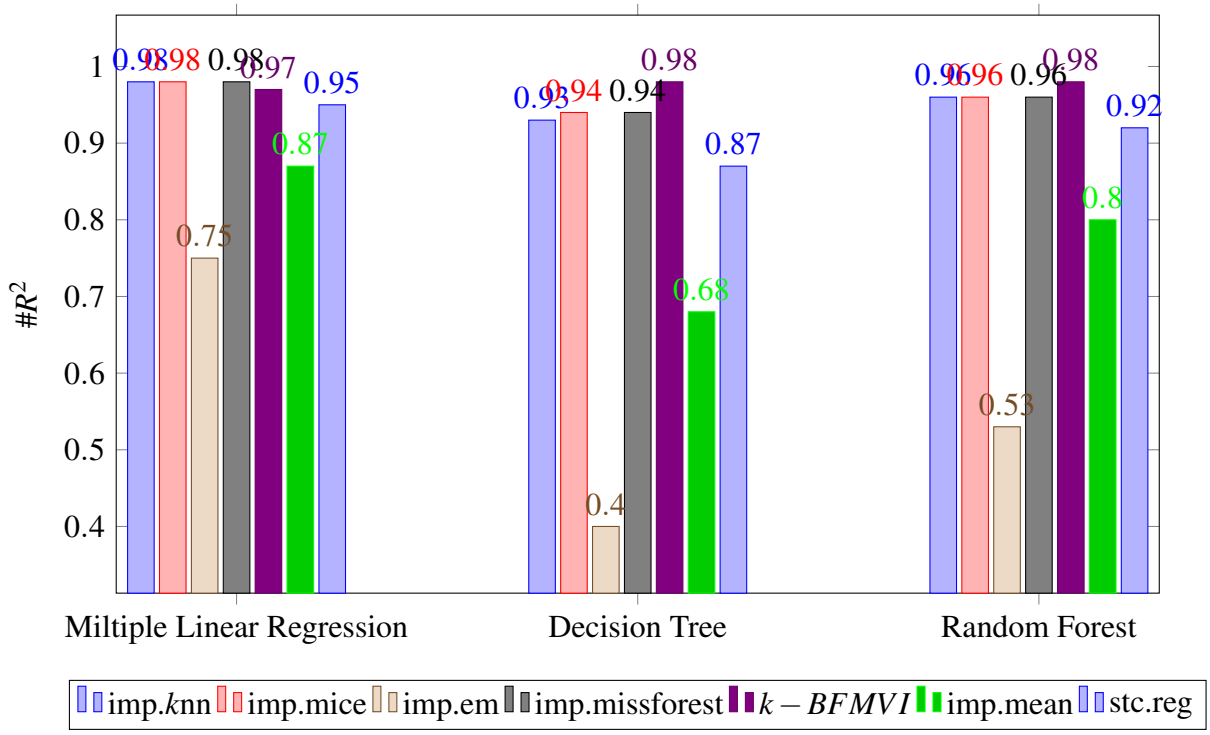

Fig. 4.2 R^2 Indicator for Supervised Learning Models Upon Imputation on MCAR Data

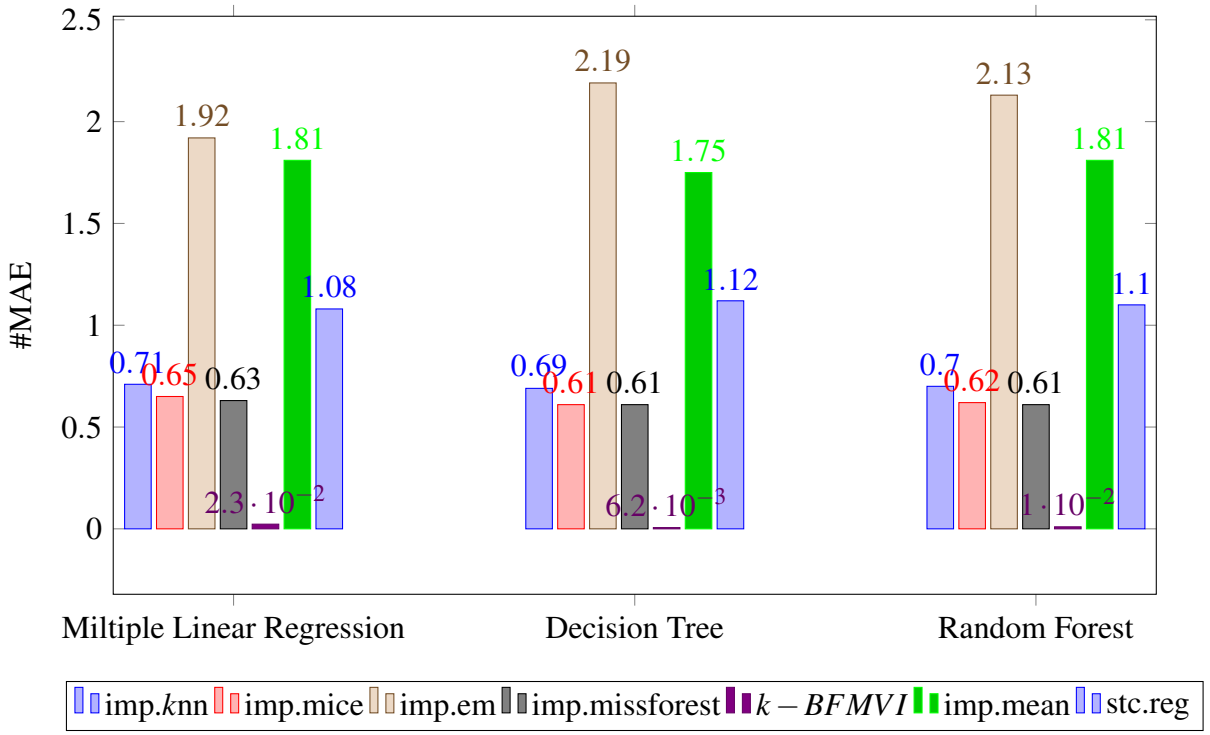

Fig. 4.3 MAE Indicator for Supervised Learning Models Upon Imputation on MCAR Data

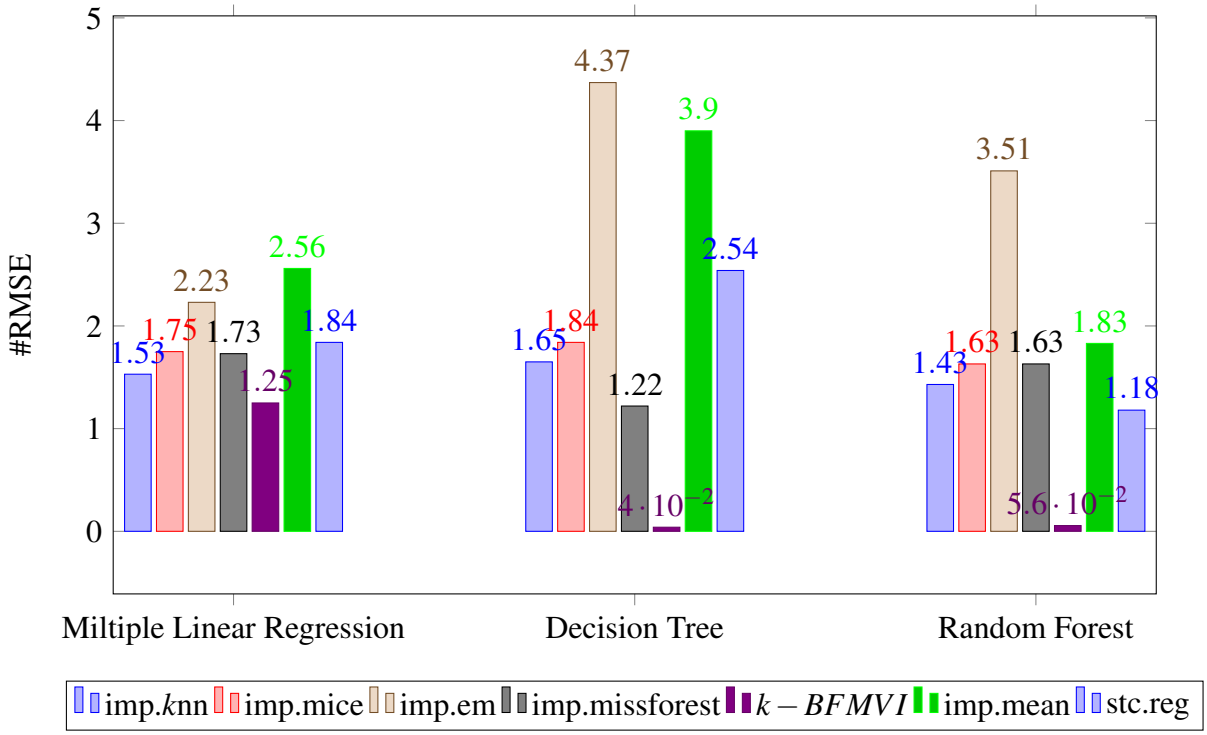

Fig. 4.4 RMSE Indicator for Supervised Learning Models Upon Imputation on MAR Data

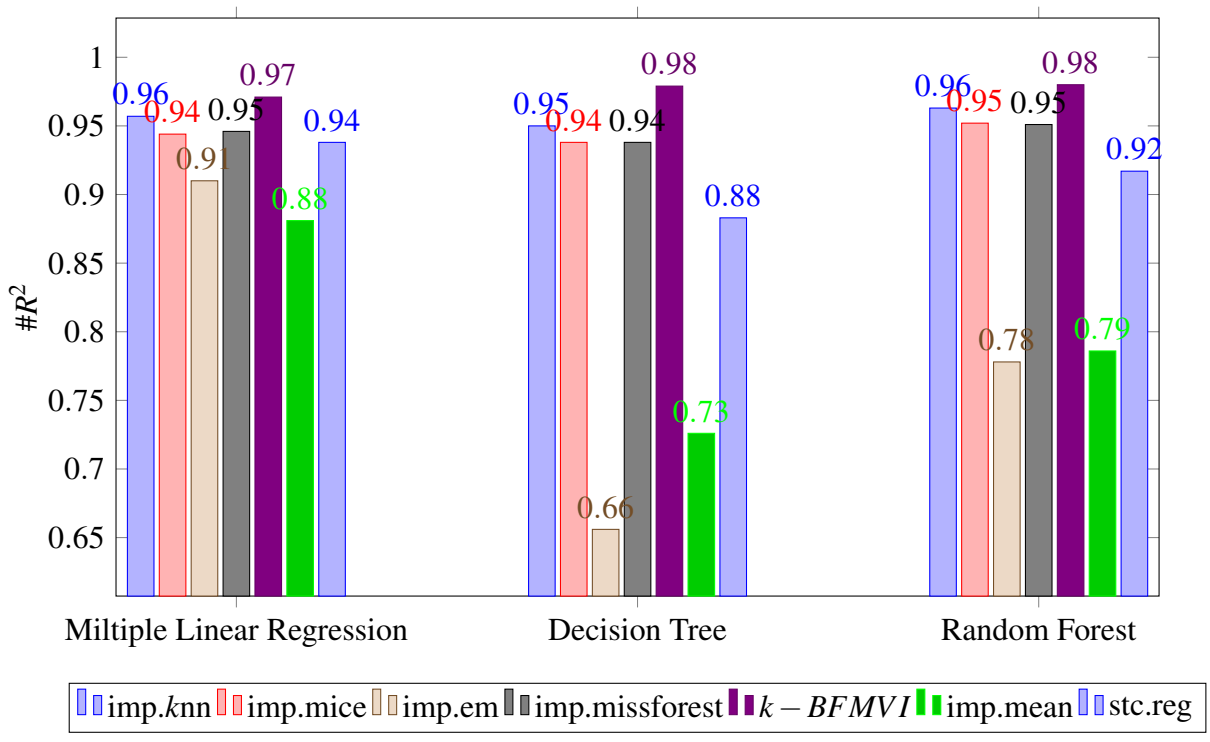

Fig. 4.5 R^2 Indicator for Supervised Learning Models Upon Imputation on MAR Data

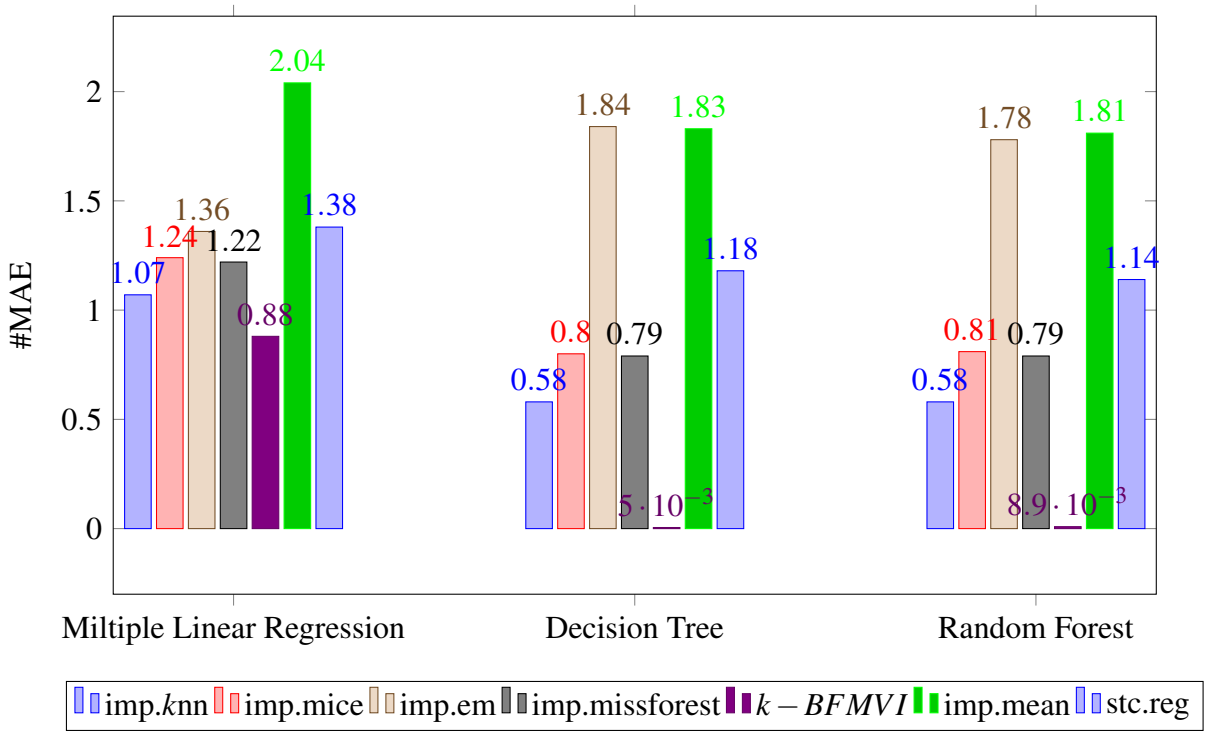

Fig. 4.6 MAE Indicator for Supervised Learning Models Upon Imputation on MAR Data

a boundary from the relationships of covariates and base the imputation on observed records that fall within the specified boundaries. In addition, this thesis comparatively demonstrates the performance gains that can be observed when the proposed method is trained on learning algorithms against benchmark techniques.

The author proposed the $k-BFMVI$ technique, which is a robust combination approach that uses cross-validation to make a choice among embedded imputation functions k-nn, missForest and Regression imputation models. The author presents evidence of $k-BFMVI$'s empirical performance and comparatively assess performance gains against benchmark methods over large scale computational tasks on real-world sensor data with $n > 6000$ observations. For all the missing data scenarios considered, $k-BFMVI$ yielded the best performance in terms of imputation accuracy for 10% to 40% missing ratio signified by the lowest RMSE and MAE score. However, a trade off can be seen between the accuracy and

complexity of the proposed technique as it attracts more computational costs due to the tasks generated by the sub-techniques embedded in the algorithm.

Furthermore, the experiments show that using imputations generated from $k-BFMVI$ where parameter values are nearest to the ground truth, results in gains on downstream regression tasks. This result suggests that for low to high missing ratio, practitioners in machine learning can benefit from adopting this model, especially for highly correlated data sets.

Chapter 5

Imputation of Missing Clinical Covariates for Classification Problems

5.1 Introduction

The application of machine learning techniques in healthcare can lead to the generation of actionable insights ranging from streamlining operations in hospitals to early detection of diseases. Statistical techniques that exploit the richness and variety of clinical data are relatively sparse, creating an avenue for further research in this area (Callahan and Shah, 2017). As vast amounts of information are available today, experts in the medical field have become reliant on machine learning techniques for performing various tasks (Obermeyer and Emanuel, 2016).

These information will be generated from numerous epidemiological and clinical sources such as claims records, data from longitudinal studies and clinical trials which over time have become invaluable assets for medical research. In most of these studies, data is gathered over time from individual subjects via repeated or continuous monitoring and assessment of both health outcomes and risk factors. For instance, longitudinal studies are used to find the correlation between levels of exposure and health effects such as chronic diseases. Originally,

these studies collect prospective data, prior to the knowledge of future events, therefore mitigating bias from the responses of participants (Caruana et al., 2015).

Another popular and valuable means of obtaining clinical data are Electronic Health Records (EHR). The widespread adoption of EHR over the years has led to the generation of massive amounts of data containing qualitative, quantitative and transactional data (Murdoch and Detsky, 2013). While primarily developed for storing patient records and enhancing administrative tasks, researchers explored secondary applications for EHRs in clinical informatics (Shickel et al., 2017).

The issue of missing data is a prevalent challenge that poses significant challenges in the interpretation and analysis of longitudinal clinical data sets Wood et al. (2004), potentially diminishing their plausibility and producing biased conclusions. The presence of missing values may cause complications in the interpretation of important insights in a study or even invalidate the entire study (Ware et al., 2012). As machine learning algorithms and statistical models rely on fully observed data sets, it is important to develop appropriate strategies to handle missing data effectively.

Classification algorithms such as Random Forests (Breiman, 2001), Classification and Regression Trees (CART) (Breiman et al., 1984), do not have built-in techniques for addressing missing values present in the training data. By ignoring instances with missing values and using only complete records in the classification algorithm, vital information may be lost in a given distribution (Little and Rubin, 2019). The presence of missing data is a major challenge for experts aiming to solve classification problems in real-life studies (Duda et al., 2012).

The aim of classification problems is to develop a classifier from a sample of training data to ensure the correct classification of new test observations. In the training set, the class membership is assumed to be stated for each observation whereas missing values may be present in corresponding features/attributes. The test data on the other hand will consist

of newly observed records with similar but unlabelled features. The goal of classification problems is to effectively assign class labels to the test data (Alpaydin, 2020). The problem formulation assumes a Missing Completely at Random (MCAR) and Missing at Random (MAR) mechanism in the training and test data set. This thesis identifies one approach to classification where instances with missing values are ignored before building a classifier. This approach can only yield effective results when the amount of missing data is relatively low. However, research has shown that adequate imputation techniques can improve classification accuracy even for a missing rate of 5% (Farhangfar et al., 2008).

5.2 Classification Frameworks with Missing Features

Most generative classifiers possess the ability to overcome the missing data problem through marginalisation, thus making them suitable for handling MAR data. Popular approaches to handling missing data for discriminative classification are complete case analysis, learning in sub-spaces and imputation. These methods can be used with any classifier that is designed for complete data sets. In this section, the author highlights some of these approaches to handling missing data for classification problems.

5.2.1 Case Deletion

In this approach, data cases with missing instances are discarded from the distribution and only records with complete observations are considered. This approach neglects useful information about features and is generally not a recommended approach for treating missing values for classification tasks (Little and Rubin, 1987). This approach may however be useful only where complete data is required at test time and the amount of missing values is minimal. The missing data scenario considered in this thesis, however requires the classification of incomplete data vectors rather than using case deletion.

5.2.2 Generative Classifiers

These classifiers model the shared distribution of features and labels. Here, features with missing values can be marginalised when data cases are being classified. Class conditional models such as Linear Discriminant Analysis and Naive Bayes classifiers can perform marginalisation efficiently and missing value treatment must occur during learning. Generative classifiers usually require assumptions to be made regarding the feature space. However, this is not required for discriminative classifiers (Marlin, 2008).

5.2.3 Classification and Imputation

The strategy the author focuses on in this thesis is imputation, which is widely used in the research community as a method for treating missing data. Imputation approaches can be classed into single and multiple imputation approaches. In section 3.2.2 of this thesis, mean imputation is described as a popular single imputation technique. In the mean imputation technique, it can be seen that an exclusive completion of a feature p can be derived by generating a single value for replacing missing observations. Multiple imputation on the other hand generates multiple plausible values for each missing instance. A principal merit of multiple imputation lies in its ability to reflect the variability in a distribution with missing values. Approximate Bayesian techniques are relatively sophisticated variations of multiple imputation such as Markov Chain Monte Carlo (MCMC) algorithm. This can be seen as an estimation to consolidating missing data over a feature space with respect to auxiliary distributions.

An effective imputation technique relies on the selection of a suitable model to effectively sample data from the input space. This is less likely to be the case for single imputation by imputing zeros. A common practice for multiple imputation is to sample several completions of missing features that have been conditioned on the complete features by fitting a Gaussian

distribution on each class. Flexible approaches to imputation for real data samples are mostly based on Gaussian mixture models (Tresp et al., 1993).

According to research, imputation poses strong advantages as they can be used in conjunction with classifiers used on complete data. However, learning multiple models for imputing missing values can attract high computational costs.

5.2.4 Classification in Sub-spaces: Network Reduction Approach

A straightforward approach to handling missing data perhaps is by learning a different classifier on the different patterns of the observed values in a data set. A study conduced by Sharpe and Solly (1995) used this approach in conjunction with neural networks to investigate the diagnosis of thyroid diseases. This is referred to as the network reduction approach. Standard discriminative classifiers can be fitted to learn each model and (Sharpe and Solly, 1995) observed that each subspace of observed features learned on a neural network classifier produced better performance compared to regression imputation based on neural network (NN) combined with an NN classifier considering all the features as inputs.

A major drawback of this approach however is that the number of missing patterns on features is exponential based on the number of features considered (Tresp et al., 1993). In the case of the data set considered by Sharpe and Solly (1995), four inputs with four missing patterns on features were considered which made the approach more feasible.

5.2.5 Classification Through Response Indicators

An alternative approach to reduced models and imputation is the augmentation of an input to a classifier with response indicators. Considering an input $\widehat{p}_n = [p_n \odot v_n, v_n]$, which can be seen as an encoding for p_n^o, where $\odot$ represents elementwise multiplication. Lets assume a decision function $f(p_n^o)$ which represents the trained classifier. In multi-layer NN, logistic regression and some kernel based classifiers, making a substitution of $\widehat{p_n}$ for p_n is the only

adjustment required. This approach was investigated jointly with SVM models in a research by conducted by Chechik et al. (2006) for structural incomplete data problems. This type of incompleteness arises when specific feature values for some data cases are not defined. This type of missingness however differs semantically from the types of missing data considered in this thesis.

5.3 Methodology

5.3.1 Formulation of the Problem

Let $P = \{P_i\}_{i=1}^{n}$ be an $n \times p$- dimensional matrix of n distinct observations having p attributes/features and V is a response variable having class labels that are influenced by P. This thesis takes into account no dependence structure between the attributes in P. Let D be an $n \times p$ matrix showing the missingness of the corresponding features of P. In practice, incomplete data is generated for a random size n of the population (P,V,D) set as the training data which was used to train the classifier

$$D = \{(P_i, V_i, D_i)\}_{i=1}^{n}, \tag{5.1}$$

where the class labels in $V_{i=1}^{n}$ are fully observed, $P_i = (P_{ij})_{j=1}^{m} = (P_{i1}, \ldots, P_{ip})$ denotes the m features of the i-th observation measured along the indicator variable $D_i = (D_{ij})_{j=1}^{m}$ where

$$D_{ij} = \begin{cases} 0, & P_{ij} \text{ is missing} \\ 1, & \text{otherwise.} \end{cases} \tag{5.2}$$

without any loss of generality in the matrix, lets assume that for each i, the observation $P_i = (P_{ij})_{j=1}^{m}$ contains m_0 categorical attributes for $j \in \{1,2,\ldots,m_0\}$ and m_1 continuous features for $j \in \{m_0+1,\ldots,m_0+m_1\}$ such that $m_0 + m_1 = m$. Let the j-th categorical

attribute contain k_j distinct values of the j-th continuous variable that represents the $(m_0 + j)$-th feature of P_i, indexed by $j \in \{1,\ldots,m_1\}$ and takes the values from a continuous set $C_j \subset \mathbb{R}$. We can map the k_j distinct values to the initial k_j natural values for each categorical features, such that $P_i \in \{1,\ldots,k_1\} \times \cdots \times \{1,\ldots,k_{po}\} \times C_1 \times \cdots \times C_{p1} \subset \mathbb{R}^m$.

Here, lets assume that the $\{(P_i,V_i)\}_{i=1}^n$ satisfies the model

$$V_i = g(P_i), \;\; i = 1,2,\ldots,n, \tag{5.3}$$

where $g(.)$ denotes an unknown function which maps a p-dimensional number (which belongs to a subspace of $\mathbb{R}^m$) to a discrete set R which represents the class labels and $V_i \in R$. The author assumes that R has m values and therefore, the classification problem is established from m classes.

The goal of any classification task is to make use of a training set $\{(P_i,V_i)\}_{i=1}^n$ to produce estimates for $g(.)$. This can be referred to as 'training' a classifier $\hat{g}(.)$. Considering a new test set of L observations, $P^{'} = \{P_i^{'}\}_{i=1}^L$, the corresponding classes $V^{'} = \{P_i^{'}\}_{i=1}^L$ are predicted by the classifier using $\hat{V}_i^{'} = \hat{g}(P_i^{'})$. It is noteworthy that missing values can also be present in the test set $V^{'}$. According to Luengo et al. (2012), the accuracy of various classification algorithms have shown improvements following the imputation of missing values in the matrix of feature P before training the classifier. This chapter proposes the $med.BFMVI$ imputation algorithm which first imputes missing values during training. The algorithm is further extended to impute the missing values present in the test set $P^{'}$.

5.3.2 Classification and Regression Trees (CART)

Here, a demonstration of how Classification and Regression Trees (CART) are exploited in longitudinal studies is presented. As previously stated, response matrix V with time-varying $n \times P$ entries as well as the demographic and time varying covariates (which are replicated

t number of times) are observed in the data matrix P with $n \times t \times p$ dimensions. Both V and p two dimensional matrices P_{ij} which represents the values of the p covariates can be categorical or continuous data, which signifies that P is a dataset with mixed data types. This thesis takes into consideration the k-th attribute of the matrix P which is populated with a series of binary splits which are represented as $I\{P_{...k} \leq c\}$. When a tree is built using the split, observation points with $\{P_{...k} \leq c\}$ are classed into one area generated by the split and subjects with $\{P_{...k} > c\}$ will fall into another area. The splitting point is considered as the internal node while the final end points are taken as the terminal nodes (Breiman et al., 1984).

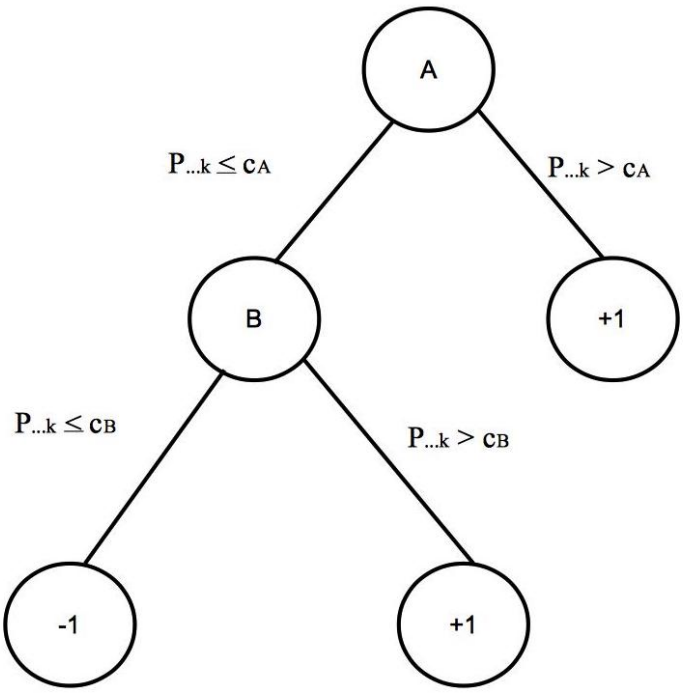

Fig. 5.1 An example of CART with two partitions.

The regions in the split data matrix P can be defined as $R_m, m = 1, \ldots, M$, where M represents the number of terminal nodes. The average outcome of the terminal node m can be given as

$$\hat{c}_m = \sum_{i=1}^{n} \sum_{j=1}^{t} \frac{V_{ij} I\{P_{ij.} \in R_m\}}{\sum_{a=1}^{n} \sum_{b=1}^{t} I\{P_{ab.} \in R_m\}'}, \quad \text{and,} \tag{5.4}$$

$f(P_{ij}) = E[V_{ij}|P_{ij}]$ can then be estimated based on

$$\hat{f}(P_{ij}) = \sum_{m=1}^{M} \hat{c}_m I\{P_{ij.} \in R_m\}. \tag{5.5}$$

Therefore, the estimated value of the new data point is equivalent to the average value of the outcomes measured across all data points of the training set over different time-points.

When building a classification tree using the greedy approach, all m variables and their potential cut points are taken into consideration at each node, then we move forward with the cut point which ensures a maximum gain where the suitable criteria is concerned. In order to choose the cut point and variable for each node, the sum of squares minimisation is performed, given as

$$\min_{k,c} \left\{ \min_{a} \sum_{i,j:P_{ij.} \in R_t(k,c)} (V_{ij} - a)^2 + \min_{b} \sum_{i,j:P_{ij.} \in R_g(k,c)} (V_{ij} - b)^2 \right\},$$

where $R_t(k,c) = \{P|P_{...k} \leq c\}$ and $R_g(k,c) = \{P|P_{...k} > c\}$. It is important to note that the values of a and b are estimated using the average of the outcomes in R_t and R_g respectively. When handling missing data problems, if $P_{...k}$ is unobserved, it is omitted from the calculation of the split points. The stopping requirement for tree growth is a terminal (minimum) node size. This means that a terminal node is expected to have at least the minimum node size. The tree growth is completed when all the terminal nodes can no longer be partitioned. This is called the maximal tree T_0.

In order to reduce over-fitting, the maximal tree can be cut back using cost-complexity pruning. Lets take T as a subtree of T_0 and the total number of nodes in the tree T is represented as $|T|$. Subtrees can be created by sequentially collapsing the non-terminal nodes that yield the smallest value of

$$\frac{1}{M}\sum_{m=1}^{M}\sum_{i,j:P_{ij}.\in R_m}(P_{ij}-\hat{c}_m)^2.$$

The tree generated within this sequence given as

$$C_\alpha(T)=\sum_{m=1}^{M}\sum_{i,j:P_{ij}.\in R_m}(P_{ij}-\hat{c}_m)^2+\alpha M,$$

which minimises the cost complexity criterion and is then chosen as the final tree. α can be chosen by minimising the cross-validated sum of squares.

5.3.3 Random Forest Based Classification

The Random Forest (RF) algorithm was first introduced by Breiman (2001) and manages to reduce the correlation of the samples constructed by the trees. These improvements can be seen by further reducing the variance of the distribution (Hastie et al., 2004). At each node of the tree, subsets were randomly chosen from the entire predictor space as seen in Algorithm 4. Different bootstrapped samples were used to grow each of the trees A. These two approaches are applied to reduce the problem of highly correlated trees. This results in a lower variance in the the estimates generated in the target variable, thereby resulting in higher classification accuracy. By sampling a subset of predictors randomly for each iteration, additional predictor variables are generated after the tree A is grown. This allows contributions from variables that are of lesser importance towards the output generated in the target variable. Lets take $(P_1,v_1),\ldots,(P_N,v_N)$ as the respective predictor and target variables of the training set. The classification problem is solved using RF by taking the frequently occurring observation class m, and the predictor values P_m as the mean predicted response $\hat{v}_m$.

In order to improve the optimal performance of the RF algorithm, a number of hyper-parameters need to be added. The total number of trees ($ntee$) determines the amount of trees

to be grown. The number of predictors m that are randomly chosen for each tree node is also determined as described in Algorithm 4 (step 4). The minimum *nodesize* which determines the depth of the tree is also added. According to research, it is crucial to determine the optimal combination of the parameter values for the random forest classifier to perform optimally (Cutler et al., 2012).

Algorithm 4 Random Forest Classification (Breiman, 2001)

Input: Data set (P,V,D) with $P \in \mathbb{R}^{n \times p}$
Output: Data set with classified labels P_s.

1: For i = 1,..., I; do
2: Construct a bootstrap set D with size N;
3: Generate a decision tree (T_a) from D based on the below procedures for each terminal node. Stop after minimum node size has been reached;
4: Choose m out of p variables at random;
5: Find the optimal split and variable based on the Gini criterion $G = \sum_{k=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk})$
6: Partition the node into two samples;
7: Return output showing the Tree (T_a) ensembles for i = 1,..., I
8: For each Tree, use majority vote to classify new observations of p:
9: Let us take $\hat{C}_b(p)$ to be the prediction of classes in the a^{th} tree of the observation p;
10: Taking the output from (9), classify p based on the rule of majority vote on all A trees where $\hat{C}_{rf}^{b}$ = Majority vote $\hat{C}_b(p)_1^A$

5.3.4 Bootstrap Aggregation (*Bagging*) Based Imputation

The *CART* approach described in section 5.3.2 is capable of capturing the relationships between explanatory variables by partitioning the predictor and estimating the values of the target variable. However, research has shown that *CART* can be noisy in terms of their predictive accuracy (Hastie et al., 2004). The *Bagging* method has been used to compensate for the variability that may be observed in classification algorithms. The variance of bagged estimates $\hat{f}_{bagg}(P)$ still increases in line with the increase in correlation between each tree. A lower correlation between estimates will therefore result in a lower variability in the final prediction (Hastie et al., 2004).

This method essentially computes an average of the prediction results obtained over bootstrapped samples. The process involved in *Bagging* includes taking each bootstrap sub-sample N and constructing A decision trees, after which predictions are made for classification problems using majority vote. As defined previously, the setback of decision tree classification is that $\hat{f}_{bagg}(P)$ could show a high variability for a number of training samples. In order to avoid these limitations, the *bagging* algorithm constructs A number of trees from a training sets at random and computes an average of predictions made in each set based on the equation below:

$$\hat{f}_{bagg}(P) = \frac{1}{A}\sum_{a=1}^{a}\hat{f}_a(P) \tag{5.6}$$

This process decreases the variance observed in the classifier even with a large tree depth. A bagged ensemble of trees have the potential forming more accurate classes compared to predictions produced from individual trees.

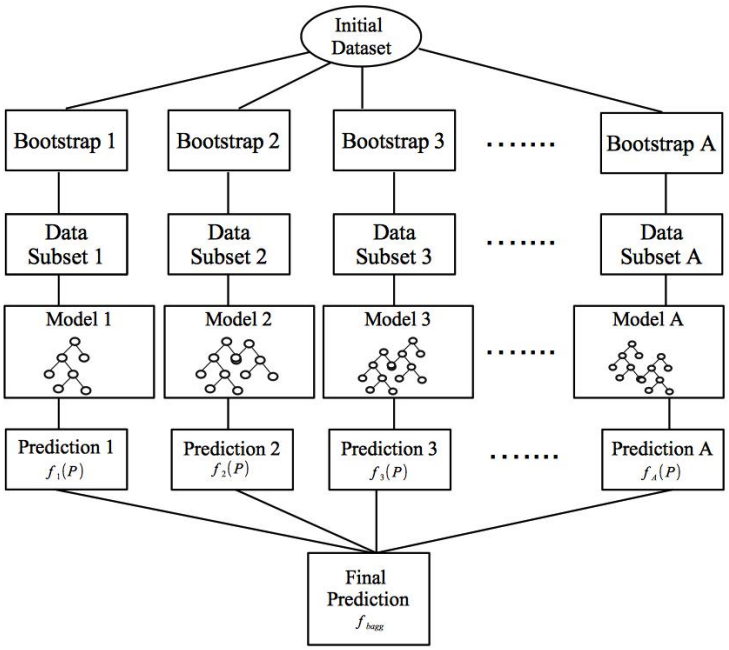

Fig. 5.2 Bootstrap Aggregation

5.3.5 Proposed Class-weighted *med.BFMVI* Algorithm

This thesis tailors the *med.BFMVI* for classification problems and encodes the target variable to create boolean variables. Considering the data with *m* categories, the class variables are simply recreated based on *m*-1 assuming that the final class variable is already known or is dependent on other identified variables.

This dependence is mathematically expressed as:

$$\sum_{i=1}^{m} p_i, \quad p_i \in \{0,1\} \tag{5.7}$$

where p_i represents the i-th binary variable and $p_i \in \{0,1\}$ simply requires that the class variables must reflect the available boolean variables. The above equation can also be represented as:

$$p_j = 1 - \sum_{i \neq j} p_i \tag{5.8}$$

which clearly shows the linear dependence that exists between other variables and the j-th variable. The interaction between the binary variables in the classification problem will always be represented as

$$p_i . p_j = 0, \quad i \neq j \tag{5.9}$$

as both variables are mutually exclusive, meaning when the first value is 0, the other value is 1.

The class weight γ_j from the j-th attribute is used, which serves as a basis for weighing the *med.BFMVI* between observations as follows

$$\text{med.}BFMVI(p_a, \ldots, p_n) = \sum_{j=1}^{m} \gamma_j \;\; \text{med.}BFMVI(p_{aj}, \ldots, p_{nj}). \tag{5.10}$$

By using the class weights, imputation of the missing continuous variables is conducted based on the sub-techniques considered in 4.4.2 and the score function in Equation 4.25.

Algorithm 5 Class-weighted *med.BFMVI* Imputation

Input: (P,V,D) with P $\subset \mathbb{R}^{n\times p}$ having missing values, V contains m class labels
Output: Imputed matrix $\widehat{P}$.

1: **Pre-processing the Data:** Lets first of all transform class variables of V using (5.8).
2: **Initialisation:** Lets consider the class labels in V and use them as a basis for splitting P in to $\{P^v\}_{v=1}^m$ considering their weights γ_j. Take each class v given P^v and pre-impute missing continuous features p_1 using mean imputation.
3: **Iterative Step:** For each missing instance i in class v, the imputed data matrix $\widehat{P}^{v,t}$ is derived using med.$BFMVI(p_a,\ldots,p_n) = \sum_{j=1}^m \gamma_j$ med.$BFMVI(p_{aj},\ldots,p_{nj})$. This process is repeated for each v using each sub-technique from 4.4.2 to obtain $\widehat{P}^t = \{\widehat{P}_{v,t}\}_{v=1}^m$.
4: **Error Calculation:** Lets use the error score function in 4.25 to determine the choice of imputation. The sub-technique with the lowest error is used to impute missing values for instances in v.
5: **Stopping Criteria:** Stop when the imputed values with the lowest RES(r) score is used to impute missing values in $\widehat{P}$

5.4 Computational Experiments with Real-world Data

In this section of the thesis, computational benchmarks are reported in order to compare robust approaches and their counterparts. The author further explores problem scenarios with impacts on the performance gains of classification approaches.

5.4.1 Experimental Setup

For the purpose of reporting the comprehensive performance of classification techniques on real-world applications, this thesis evaluates the accuracy of imputation techniques on classification problems of real-world clinical data reporting Cardiovascular Diseases (CVD) among a range of participants obtained from the UCI Machine Learning Repository (Frank, 2010). CVD are mostly identified as conditions that involve the blockage of blood vessels

thereby causing ischemic heart disease (IHD) (angina, myocardial infraction) or stroke (Almas, 2018). These conditions prevent the flow of blood to the brain or heart. To obtain the binary classification, various health and lifestyle factors of participants were considered in the dataset to identify plausible diagnosis of cardiovascular diseases.

For the purpose of experiments, missing data is generated at different rates ranging from 10% - 40% for different missing data mechanisms. The fully observed data generated from CCA was taken as the ground truth. The different imputation techniques were applied on the range of datasets generated and their performances were compared against all the techniques embedded in the optimised imputation algorithm.

The individual methods used in the comparison are:

1. imp.*k*nn: This is a single-step method which uses the nearest neighbours to perform imputation based on the euclidean distance. The membership neighbours are expected to have no missing values present in the imputed feature. If other coordinates are missing, an averaged distance can be used. This was implemented using $scikit-learn$ package in Python.

2. imp.mice: Missing data estimation was also carried out by running a multiple linear regression model using MICE implemented in Python using $Bayesian\ Iterative-Imputer$.

3. imp.missforest: This is a non-parametric method based on the random forest algorithm, implemented using the *missingpy* library in Python.

4. imp.mean: The mean of the distribution was also simply computed and this value was used to replace missing instances. The imp.mean technique was implemented using Python $SimpleImputer$ library.

5. stc.reg: A stochastic regression model was run on the data and the predicted values were used in place of the missing instances.

6. *med.BFMVI*: The sub-techniques below provide a low level description of the proposed algorithm and follows a warm starts optimisation approach where non-missing values of a related feature are used for imputation. As previously mentioned, the imputation with the lowest RES(r) as seen in equation 4.25 is selected.

- k-NN based ($med.BFMVI_{knn}$): This sub-technique is used to solve the optimal problem of finding the k-nearest neighbours before imputation. The class dependencies of P were used at the initial stage to improve the quality of the imputation. The algorithm was run on the dataset of size n = 10,000.

- *Regression* based ($med.BFMVI_{reg}$): This method builds multiple linear regression models for each weighed class. The resulting predictions are used to replaces only the instances with missing values.

- *RandomForest* based ($med.BFMVI_{missForest}$): This is method is based on the random forest algorithm described in 4.21. This technique takes each class weight and builds a random forest which generates plausible estimates which is in turn used to replace missing values.

Missing Data Pattern

Because different missing data mechanisms affect the quality of imputation, simulations were conducted considering two missing data mechanisms: missing at random (MAR) and missing completely at random (MCAR). These statistical assumptions are summarised in table 5.1. In order to generate the MCAR mechanism, a subset of the records in P was sampled at random, assuming that each entry has equal probability of being chosen. The MAR mechanism was generated by sampling the entire dataset and modelling the missing data probability for the target variable. For instance, if $R_i = 1$ the the corresponding value of P is deleted.

Table 5.1 Missing data mechanisms used for the generation of missing data M in the data set P. lets take f to be the density of the missing data pattern. P^{miss} and P^{obs} represent the missing and observed data respectively.

Missing Data Mechanism	Statistical Assumption
Missing at Random (MAR)	$f(M\|P^{obs},P^{miss}) = f(M)$
Missing Completely at Random (MCAR)	$f(M\|P^{obs},P^{miss}) = f(M\|P^{obs})$
Not Missing at Random (NMAR)	$f(M\|P^{obs},P^{miss})$ is a function of P^{miss}

5.4.2 Results

The imputation methods were tested on real-world clinical data obtained from the UCI Machine Learning Repository. This data contains $n = 10,000$ observations and $p = 11$ dimensions. Next, the results show that the quality of imputation produced from the $med.BFMVI$ sub-methods is higher compared to other benchmark methods, which further leads to an improvement in the performance of downstream classification tasks. This thesis evaluates the performance of these methods considering different missing rates for MACR and MAR conditions.

Imputation Accuracy

The imputation accuracy for each technique was evaluated, assuming the MCAR condition. Among all the methods considered, it can be seen that at least one of the $med.BFMVI$ techniques record the lowest RMSE and MAE scores for 10% - 40% missing rate indicating strong performance, followed by the imp.mice technique which is closely followed by *missForest* outside the sub-methods embedded in the $med.BFMVI$ technique. Comparatively, the $stc.reg$ technique showed the weakest performance for the MCAR condition, followed by the benchmark *knn* technique and the *knn* based sub-technique in the $med.BFMVI$ method.

The experiments were repeated for the MAR condition. Comparatively, for all the missing data ratios, the proposed method still shows the best performance in terms of imputation accuracy with the lowest RMSE and MAE for all missing data ratios. Among the benchmark

methods *stc.reg* still shows the weakest imputation performance. It can be noted that the benchmark *imp.mice* approach performs well when 10% - 20% of the data is MAR. However, for higher missing rates 30% - 40%, the *regression* approach shows the weakest RMSE score.

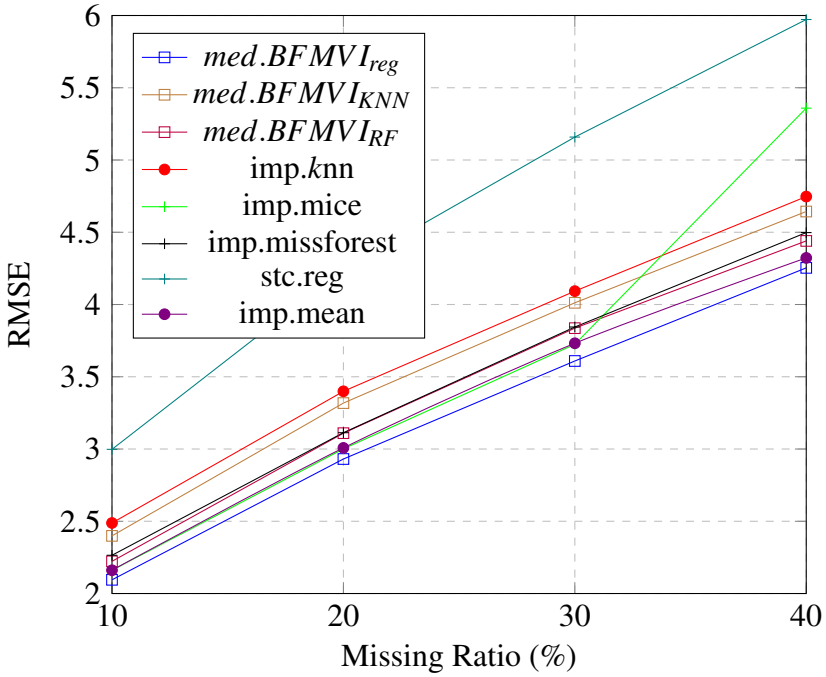

Fig. 5.3 RMSE of Imputation Algorithms for MCAR Data

Computational Complexity

Next, the computational complexity of imputation methods were compared, showing the time required to complete a cycle of imputation for the dataset with $n = 10000$ observations across each identified missingness pattern. Simulations were still conducted on a machine having an Intel Core 2 Duo (3.06 GHz) processor which is limited to 8 GB RAM. Results can be seen in Table 5.2 below.

Among the *med.BFMVI* methods, the regression based imputation scales very well considering the sample size n and dimension p for both MCAR and MAR mechanisms. Despite its imputation quality, the random forest based imputation performs relatively poor

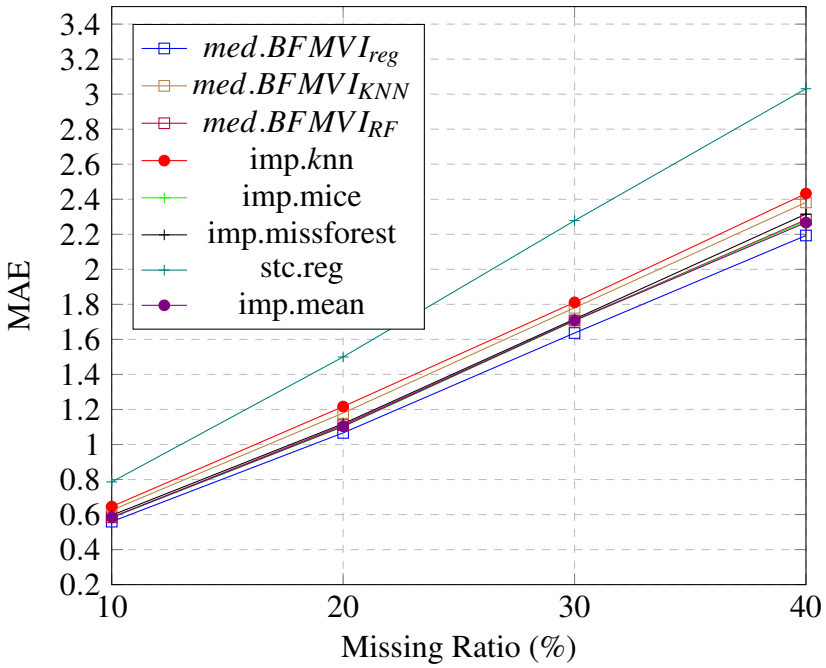

Fig. 5.4 MAE of Imputation Algorithms for MCAR Data

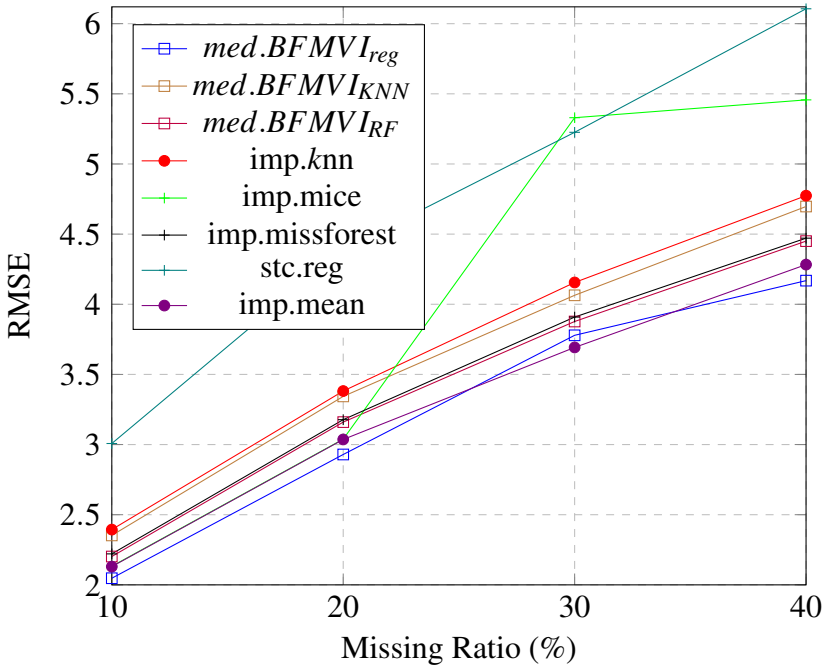

Fig. 5.5 RMSE of Imputation Algorithms for MAR Data

when compared to the other sub-techniques. Among the benchmark methods, *imp.knn* imputation performs poorly for both MCAR and MAR conditions.

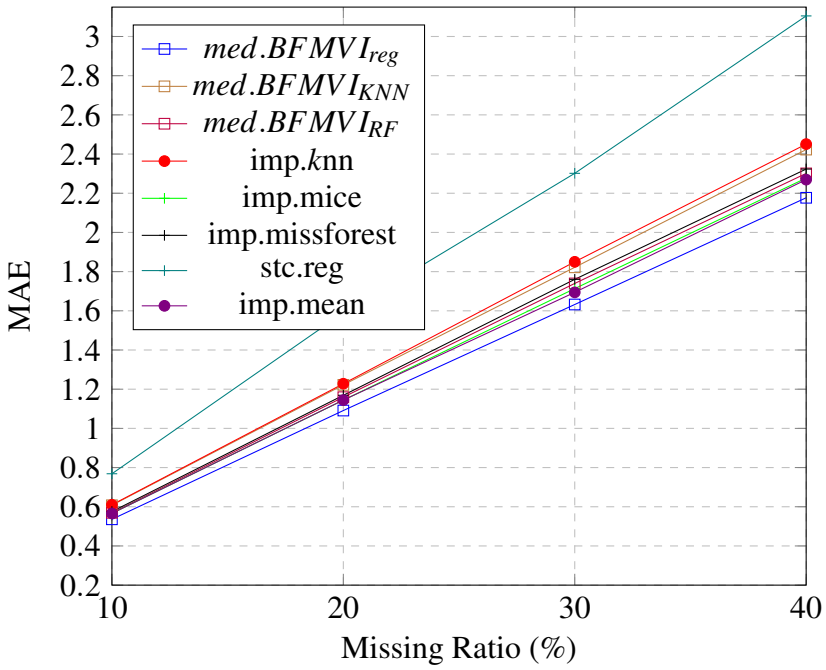

Fig. 5.6 MAE of Imputation Algorithms for MAR Data

Table 5.2 Average computational complexity for benchmark and *med.BFMVI* imputation techniques at 30% Missing Rate

	Time (s)						
Missing Pattern	imp.*k*nn	imp.mice	imp.missforest	$med.BFMVI_{reg}$	$med.BFMVI_{knn}$	$med.BFMVI_{rf}$	imp.mean
MCAR	6.07	0.76	3.36	0.13	2.37	4.06	0.93
MAR	6.12	0.73	3.43	0.14	2.26	3.31	0.70

Imputation Performance on Downstream Classification Tasks

In this section, the performance of machine learning classification algorithms trained on the range of imputed data is assessed. The challenge of classification tasks in completely observed data also differs widely across data sets.

In Table 5.3, the effect of the imputation methods on the accuracy of downstream classification tasks for MAR and MCAR scenarios are presented. Each benchmark methods and each individual *med*.BFMVI method were trained on classification problems to highlight

performance gains across each imputation technique. Simulations were further conducted using 30% missing data rate.

Similarly, when trained on downstream classification algorithms, it can be observed that $med.BFMVI$ shows comparative performance against benchmark techniques. As seen in Table 5.3, imputation done on MCAR scenario shows performance gains on all imputed data as compared to the unimputed data. Overall, from the sub-technique of the proposed approach, the $med.BFMVI_{reg}$ shows the best performance when trained on the classification algorithms with an accuracy of 89.7% when trained on the random forest classifier. An ensemble meta-estimator called bootstrap aggregation was used to train subsets of the imputed data and aggregate the individual predictions using voting technique. This led to a 1.32% increase in classification accuracy between the chosen sub-technique from $med.BFMVI$ and unimputed data. Similar response can be observed in the MAR scenario with $med.BFMVI_{reg}$ showing a bagged classification accuracy of 92.4%, showing a 1.98% gain when compared to a scenario where the classifier is trained on the unimputed data set.

Table 5.3 Classification Accuracy (%) of CVD Data at 30%

	MCAR			MAR		
Approach	CART	RF	Bootstrap Aggregation	CART	RF	Bootstrap Aggregation
No Imputation	85.2	88.1	90.9	84.9	88.3	90.6
imp.*k*nn	85.4	88.3	91.0	85.3	88.4	90.8
imp.mice	85.3	88.2	91.3	80.5	88.3	90.9
imp.missforest	85.0	88.1	91.2	84.8	88.4	91.1
imp.mean	85.3	87.8	90.7	84.6	88.1	90.7
stc.reg	77.7	83.1	87.6	77.1	82.0	86.9
$med.BFMVI_{reg}$	87.6	89.7	92.1	87.7	90.0	92.4
$med.BFMVI_{knn}$	85.6	88.6	91.0	85.7	88.4	90.3
$med.BFMVI_{rf}$	84.8	88.0	90.5	85.0	88.0	90.2

5.4.3 Discussion

One of the main contributions of this chapter is the formulation of the proposed imputation approach that is capable of estimating missing values by exploiting the pre-defined class boundaries of a target variable and choosing the best fit objective function from pre-defined sub-techniques. This algorithm first isolates the parameters of a given category and reproduces a distribution with closely tied membership covariates. Considering real-life missing data scenarios where there is no reference to ground truth, this thesis proposes a reverse error score function, embedded in the imputation algorithm to identify the most plausible technique from a selection of pre-defined imputation sub-techniques.

This chapter presents the performance of each sub-technique in the proposed $med.BFMVI$ and weighs their individual accuracy against benchmark techniques. It can be seen that the proposed technique empirically out performs benchmark techniques in terms of imputation accuracy for both MCAR and MAR conditions at 10% to 40% missing data by showing the lowest RMSE and MAE scores in the $med.BFMVI_{reg}$ sub-technique.

Furthermore, the experiments show that downstream classification results produced when trained on $med.BFMVI$ imputation technique presents the best performance with an accuracy of 92.1% and 92.4% for both MCAR and MAR scenarios respectively when bagged an ensemble of classifiers. This technique is particularly useful for clinical researchers aiming to solve classification problems where low to high rates of missing data occurs.

Chapter 6

Privacy Preserving Approach to Missing Data Imputation for Distributed Systems

6.1 Introduction

Distributed Health Data Networks (DHDNs) is a key domain that has experienced significant growth in the routine collection of data from various sources. For instance, data can be collected and stored in genome sequencing centres, patient medical databases can be created and maintained using electronic health records (EHRs) and mobile health data are largely collected using wearable devices. There has been an increase in the need for developing advanced techniques that are capable of processing the large amounts of data collected in such networks (Deng et al., 2020).

Many existing research have placed a focal point on the centralization of data in big data analysis, which follows the pattern of parallel high performance computing. However, as devices have become more intelligent, data transmission and analysis have become computationally expensive in addition to the privacy and security constraints that they pose (Stolpe, 2016). For instance, sensitive and restricted data may become more sensitive after being pooled into a central repository, such as in datasets showing clinical diagnosis

where combining patients' demographics may lead to a higher risk of disclosing personal information. Such constrained scenarios require decentralised approaches to mitigate the costs associated with the analysis of big data.

As data privacy and security have become paramount during data transmission and processing, recent state of the art investigations have shown the merits of using decentralized tamper-proof transactional databases in ensuring highly secure solutions for IoT applications such as Blockchain (Uddin et al., 2021). This chapter introduces decentralised technologies for preserving privacy and security in the analysis of IoT data. First, a general overview of Blockchain technology is presented and IOTA technology is presented as a more scalable application platform for secure and private missing data recovery. In particular, this chapter provides a detailed exploration of the IOTA Tangle and its application in the case of missing data recovery.

6.2 An overview of Blockchain Technology

Blockchain technology encompasses diverse and complex techniques including databases, computing and computing networks, cryptography, security and privacy. This technology aims to create and manage a distributed database made up of a chain of connected blocks, hence the name blockchain (Truong et al., 2019). Each block in a BC network carries a list of transactions arranged in a Merkle tree and is connected to the previous block through a cryptographic block hash. For business logic, BC serves as a distributed ledger maintained by participating nodes in a decentralised peer-to-peer network. The structure of a BC network prevents data modification as this will require negating the hashes generated and replicated in the previous blocks in the entire network, causing a break in the consensus between them. Therefore, this ensures data privacy and security as attacks on a BC network will require access to over 50% of participating nodes, which will be a challenge (Nakamoto, 2008).

Blockchain is underpinned by a consensus protocol which works to ensure that all nodes in a trustless network are in agreement before a new transaction can be added to the BC, therefore synchronising the network to ensure that a unique and consistent chain is maintained (Wang et al., 2019).

6.2.1 Consensus

Any type of distributed system incorporates a set of operations that are responsible for carrying out various tasks, and for the system to function correctly, each set of operations will be required to work collectively. This process is known as consensus. Processes in distributed networks make use of consensus protocols to coordinate the tasks that are performed on the ledger (Cachin et al., 2011). From a distributed perspective, achieving consensus among nodes is a challenging task as communication between nodes may be blocked, delayed or completely fail due to latency in the network. This inspired various research into consensus problems (Cachin et al., (2011); Castro et al., (1999)), and various fault model abstractions have been proposed in literature for distributed systems.

6.2.2 Consensus protocols

As blockchain is essentially a distributed system, issues with consensus is a fundamental problem that must be solved by blockchain protocols. As a matter of fact, there must be consensus among participants in a blockchain network on the current state of the ledger to successfully coordinate the processes on the ledger (Sagirlar, 2018). Different methodologies exist today for achieving consensus in a BC network. The following sections provide a detailed description of some related consensus protocols

Proof of Work (PoW) protocol

Proof of work protocols aim to prove the authenticity of performed operations by making computations which is mostly hardware and energy intensive (Back, 2002). In addition to that, PoW-consensus based BC networks such as Bitcoin Nakamoto (2008), requires block generators to solve a cryptographic puzzle before a valid block can be generated. The main idea behind the PoW mechanism is as follows; once a new block is formed, the block generator attaches a nonce to the block and extracts the hash value from that block. If the value of the hash satisfies a certain threshold, the block will be sealed and published to the BC network. The process of generating a block is called *mining* and block generators are referred to as *miners* (Sagirlar, 2018). In a PoW mechanism, confirming the validity of a generated block is generally less computationally intensive than generating a new block. Overall, the merit of the PoW mechanism is the prevention of instant block generation. This helps in reducing conflicts on the network such as sybil attacks and double spending.

Byzantine Fault Tolerant (BFT) protocol

To better understand this protocol, the byzantine process is first of all described. Byzantine processes are faulty or malicious operations that randomly fail from their desired tasks or algorithm. For instance, a byzantine process may fail to stick to the protocol they are assigned to, they may turn down connection requests, they may randomly drop messages/propagate incorrect information or they may stop responding completely and so on. A BFT distributed system must be designed to achieve consensus and operate effectively considering the presence of arbitrary processes that may lead to such errors as described in the byzantine process (Cachin et al., 2011).

6.2.3 State of the Art of Blockchain for M2M Economy

The Machine-to-Machine (M2M) economy is considered to be the next phase in technological evolution and the revolution of industry 4.0. With blockchain enabling security and privacy in data transmission, its potential has been studied widely for its use in the M2M economy (Mehrwald et al., 2019). However, state of the art challenges in the blockchain ecosystem has hindered its widespread acceptance due to a variety of issues such as; low TPS, storage management, low transaction confirmation rate per second, scalability issues and heavy reliance on wallets (Danzi et al., (2019); Manogaran et al., (2020)). Issues with interoperability remains one of the biggest issues with the blockchain architecture. Hence, the proposed architecture is built on the classical IOTA protocol, which provides more scalable solutions to IoT applications. In the following section, this thesis thoroughly explains the features and architecture of the IOTA platform.

6.3 An Overview of the IOTA Platform

IOTA was first introduced in 2015 and overcomes the issues with scalability faced in blockchain platforms by replacing the use of chained blocks for storing transactions with a Directed Acyclic Graph (DAG) called the Tangle (Lamtzidis and Gialelis, 2018). The tip of the Tangle consists of transactions that have been issued in the network. Before a node can attach a transaction to the network, it first of all has to verify 2 previous transactions as shown in Figure 6.1.

A Tip Selection Algorithm (TSA) is used as a reference rule by most nodes to randomly select sites where new/incoming transactions will be attached. The attachment sites, known as "Tips", are recent transactions attached to the tangle that have not been referenced by a newer/previous transaction and is therefore "unconfirmed" (Lamtzidis and Gialelis, 2018).

6.3.1 Architecture and Components of the IOTA Tangle

The architecture of the Tangle comprises of several components and layers such as nodes, transactions, network types and APIs. The building block of transactions over the IOTA Tangle comprises a transaction value, tag, transaction hash, confirmation status, address, nonce, bundle and the address of the originating transaction (Abdullah et al., 2022).

A node in the IOTA Tangle is any computer/user that generates and sends transactions. New transactions create an edge that have the function of validating two previous transactions as seen in figure 6.1, and transactions are directly or indirectly connected to the originating node. Each node validates previous transactions by solving a cryptographic puzzle. The central node (*Coordinator*) in the Tangle works to select tips for approval. When a tip is approved, the transaction is said to be *Confirmed*. The resultant outcome of this process after tips have been approved is called *Milestone* (Bartolomeu et al., 2018). Transactions in the

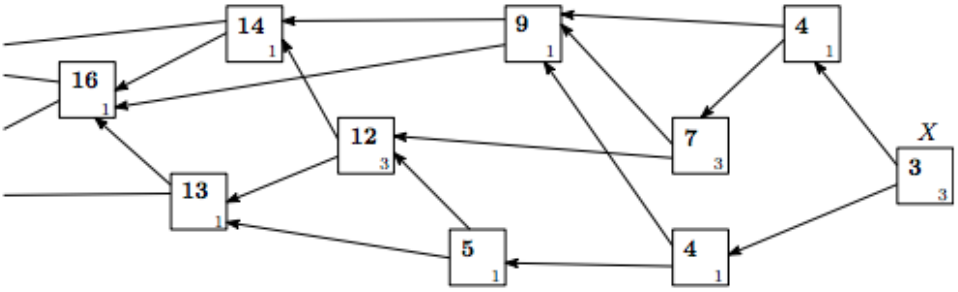

Fig. 6.1 The Tangle (Popov, 2018): The incoming transaction X directly references transactions "4" and "4". The new transaction X will also indirectly reference all other transactions that are directly or indirectly referenced by the two transactions

.

Tangle have the capability of carrying messages and IOTA Tokens. The tangle uses a ternary system instead of the conventional binary system. This system can be in equilibrium (-1, 0, 1) or it can be unbalanced (0, 1, 2). A balanced ternary system is mostly utilised in the tangle, which uses trits rather than bits. Therefore, the number of possible combinations as seen in the works of Abdullah et al., (2022) is given as:

$$1\ Byte = 2^8 = 256\ combinations;\ 1\ Tryte = 3^3 = 27\ combinations$$

A Tryte can have a maximum value of 13 with 27 distinct combinations. The ASCII sequence is used to represent tryte alphabets with a range from A-Z, beginning with number 9 (Abdullah et al., 2022).

Nodes in the IOTA Tangle can send and fetch data using a protocol called Masked Authenticated Messaging (MAM) (Handy, 2017). When a transaction is sent using MAM protocol, a channel is automatically created, and nodes that are subscribed to the MAM channel will receive transactions that are sent.

Due to the fact that IOTA is a decentralised platform, nodes can forward transactions to any participating node. Therefore, there is a risk of malicious nodes hijacking the channel. To solve this problem, a message signing is used to encrypt data that are sent across the channel based on the MAM protocol (Abdullah et al., 2022).

6.4 Distributed Health Networks: A Privacy Preserving Approach to Missing Data Recovery

This section contributes to the works of Jagannathan and Wright (2008) by exploiting the IOTA Tangle as a privacy-preserving solution to missing data recovery in IoT settings. An imputation framework is presented for horizontally partitioned databases across two sites, meaning that users across each site hold similar set of features collected at different intervals. The imputation framework is tested based on two conditions where data is evenly and unevenly distributed across each site as seen in Table 6.1. The aim of the privacy preserving approach is to facilitate missing data recovery without the need to share subject level information across each site, thereby preserving the information held within each data site as seen in Figure 6.2.

Table 6.1 Distribution Samples

Type	K	N	$n^{(1)}$	$n^{(2)}$
-	1	10000	10000	-
Even Distribution	2	10000	5000	5000
Uneven Distribution	2	10000	4500	5500

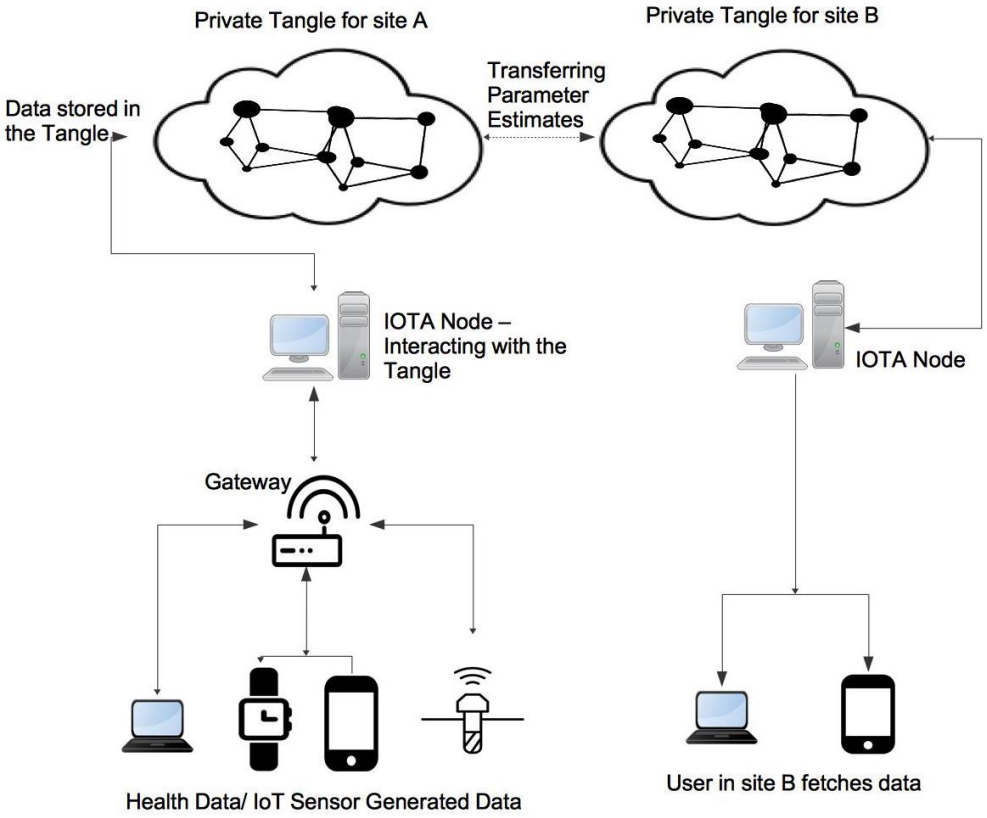

Fig. 6.2 Proposed Framework for Missing Data Recovery over the IOTA Tangle

6.4.1 Formulation of the Problem

The missing data problem in this case assumes an instance where there are missing values to be learned on a single attribute. In the distributed setting considered in this thesis, two sites hold subsets of a universal sample, where $\lambda \in S_A$ indicates the record stored in site 1 and similarly, $\alpha \in S_B$ indicates the records stored in site 2. Both sites have a database which is horizontally partitioned where $S_A = (s_1, \ldots, s_n)$ and $S_B = (s_{n+1}, \ldots, s_l)$ formed over a

universal set $\{D_1,\ldots,D_m\}\cup Q$. The attribute Q will have the missing values and is therefore referred to as the class attribute. Overall, S_A and S_B will form a universal database $S = S_A \cup S_B$ with no missing values present. A missing instance $I \in S_B$ is identified in $(s_{n+1},\ldots,s_l)$ for the attribute Q and a user in site 2 wishes to recover the missing instance $I(Q)$ based on $S = S_A \cup S_B$ without the need to reveal subject level information between sites.

This thesis further proposes a seamless approach to missing data recovery called Average Site Mixtures (AvSM) to achieve fast recovery of missing values using the IOTA Tangle.

6.4.2 Imputation Based on Average Site Mixtures (AvSM)

Suppose the missing attribute Q_1 is continuous and assumes a normal distribution set by V and other related covariates. Without sharing subject level information, each site first of all measures their model parameters using the available data in each site only. The publishing site then sends their model parameters to the subscribing site, which then combines both parameters to obtain a global estimate. Lets assume $\widehat{Q}^{(A)}$ to be the parameter estimate from S_A. The AvSM is computed by:

$$\widehat{Q}_{AvSM} = \frac{1}{k}\sum_{k} w_{Bk}\widehat{Q}^{(A)} \tag{6.1}$$

where k is the total number of sites and w_{Bk} is the current estimate for S_B. This approach is communication efficient as it requires less communication between sites and does not require the sharing of subject level information. The process involved in the AvSM approach is detailed in Algorithm 6.

6.4.3 Experimental Setup

The AvSM approach was implemented using the IOTA hornet.lib v0.6.0 (see Figure 6.3) to establish communication between the two sites having the horizontally partitioned data. The

Algorithm 6 Average site Mixture Imputation (AvSM)

Input: $S_B = (s_{n+1}, \ldots, s_l)$
Output: Global parameter estimate $\widehat{Q}_{AvSM}$.

```
Initialise request on IOTA Tangle
for k = 1,..., K do
    Request estimate Q̂^(A) at site S_A
    Fetch Q̂^(A) computed at site S_A
end for
Compute parameter estimates for local site
for i = 1,...,N do
    Compute local estimate w_B = (1/N) Σ_{i=1}^{N} Q
    Compute global parameter estimate:
    Q̂_AvSM = (1/k) Σ_k w_Bk Q̂^(A)
end for
Return Q̂_AvSM
```

network was simulated using the IOTA devnet which has been upgraded to *chrysalis* [1] to interact with both sites using secure transactions sent over the tangle. Simulations were ran using the PyOTA library, which is compatible with Python 3.6 and 3.7 to interact with the full node to send and fetch transactions over the network.

For the purpose of the experimental analysis, the clinical data set introduced in 5.4.1 was used in the simulations with the missing rate based at 30%. As missing data recovery is done at the subscribing node, the missing data recovery techniques were based on fast imputation approaches where only the required target attribute is required and not the covariates present in each site. This approach is also well suited for time series analysis. The following methods are thereby employed for missing data recovery in addition to previous methods used;

1. imp.mode: The author used the most common value approach for missing data recovery where the mode of the missing feature in the requesting site was used to replace missing values. This was implemented using Python $SimpleImputer$.

[1] https://api.lb-0.h.chrysalis-devnet.iota.cafe/

2. imp.interpolate: Imputation was also simulated using linear interpolation. This approach assumes a straight line and performs imputation in increasing order from the previous observed values.

3. imp.LOCF: Missing values were also simply replaced using the last observation carried forward approach where the last observed score is used to replace missing instances.

4. imp.LOCB: This approach is similar to LOCF but imputes missing values using a newer value in place of the previous value that is missing.

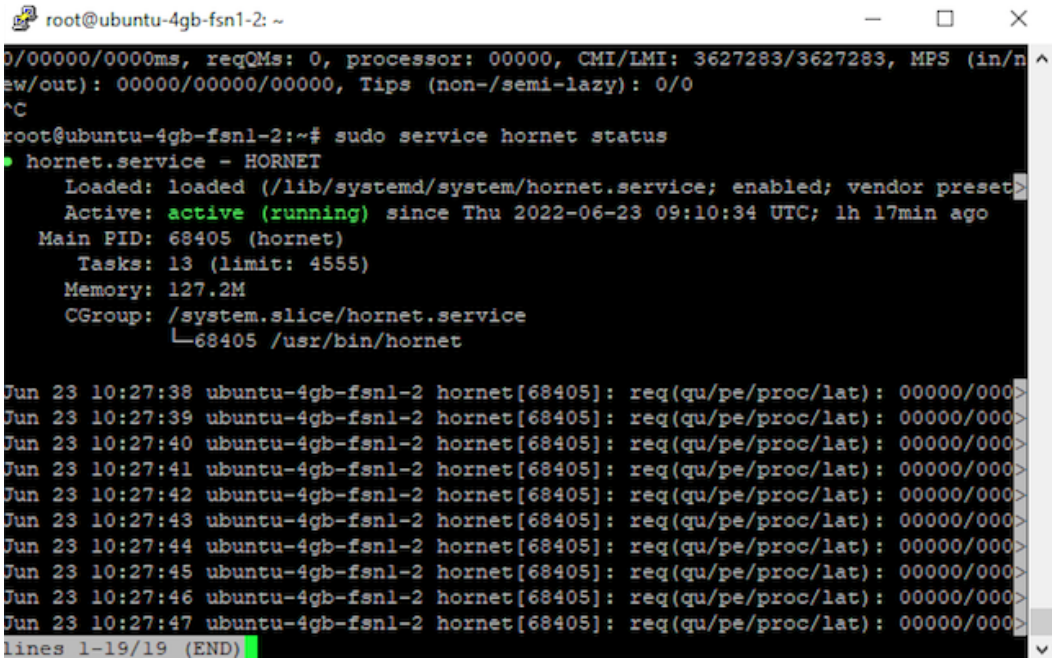

Fig. 6.3 Hornet Node Configuration

6.4.4 Results

The performance of each imputation technique was evaluated considering different data distributions. The results show that the AvSM approach presents the best imputation accuracy among benchmark fast imputation techniques with an RMSE score of 3.773 and 3.598 for even and uneven site distributions respectively. Conversely, the LOCF approach performed

poorly for both distribution samples with the highest RMSE and MAE score, indicating low imputation accuracy.

Table 6.2 Performance of Imputation Techniques at 30% Missing Rate

Type	Method	rMSE	mAE
Even Distribution (E) (5000,5000)	imp.*k*NN	4.105718	1.863427
	imp.interpolate	4.592486	2.095954
	AvSM	3.772927	1.763069
	imp.mode	4.543641	1.992402
	imp.LOCF	5.314654	2.396321
	imp.LOCB	5.20899	2.323735
Unven Distribution (U) (5500,4500)	imp.*k*NN	3.923313	1.710485
	imp.interpolate	4.379202	1.905794
	AvSM	3.59759	1.602926
	imp.mode	3.608374	1.585091
	imp.LOCF	5.067831	2.178909
	imp.LOCB	4.967073	2.112909

6.5 A Privacy Aware Distributed Intelligence Framework (PADI)

The IOTA platform has demonstrated the potential for providing privacy-protecting solutions for the recovery of missing data. A proposed framework, called the Privacy Aware Distributed Intelligence (PADI), utilizes IOTA's Masked Authenticated Messaging (MAM) protocol to secure data and adheres to the restricted access theory outlined in section 2.2.1. A system architecture for the PADI approach is presented, detailing all necessary components such as IoT devices, transaction data flow, a Node JS server with MAM, a gateway, and a PoW computation server. The IoT devices are responsible for transmitting transaction data using the MAM client. The gateway is connected to the internet and sends transaction data to a

server that operates the Node JS Masked Authenticated Messaging (MAM) application. As shown in Figure 6.4, the Node JS MAM sends the transaction data to the IOTA Tangle. For instance, the dashed lines representing the transaction data flow from IoT devices indicate the use of MAM to transmit data to the IOTA Tangle. The PoW-enabled server is assigned to carry out intensive computations on behalf of IoT devices with limited capabilities.

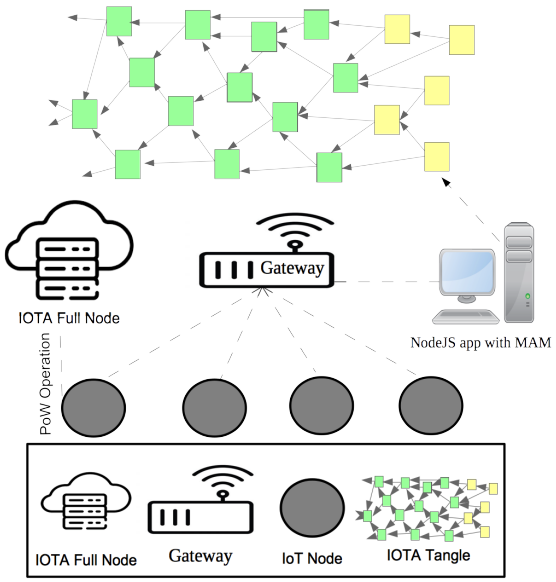

Fig. 6.4 The Proposed Privacy-Aware Distributed Intelligence Approach(PADI)

The diagram of the proposed PADI framework is illustrated in Fig. 6.5. The process begins with the initialization of the MAM state using the primary Tangle provider. After the MAM state is initialized, it is linked with the node for PoW computation, which connects it with the PoW provider. The channel mode is then set to restricted on the MAM state. Once this step is done, the payload is generated and ready to be transferred to the main Tangle. The PoW is then executed by the PoW provider. Once the PoW is finished, the payload will be linked to the main Tangle. To allow users to access records, the root ID and the appropriate secret key must be provided in the form. Thus, if the correct secret key is supplied, access is granted to that user, otherwise access is denied.

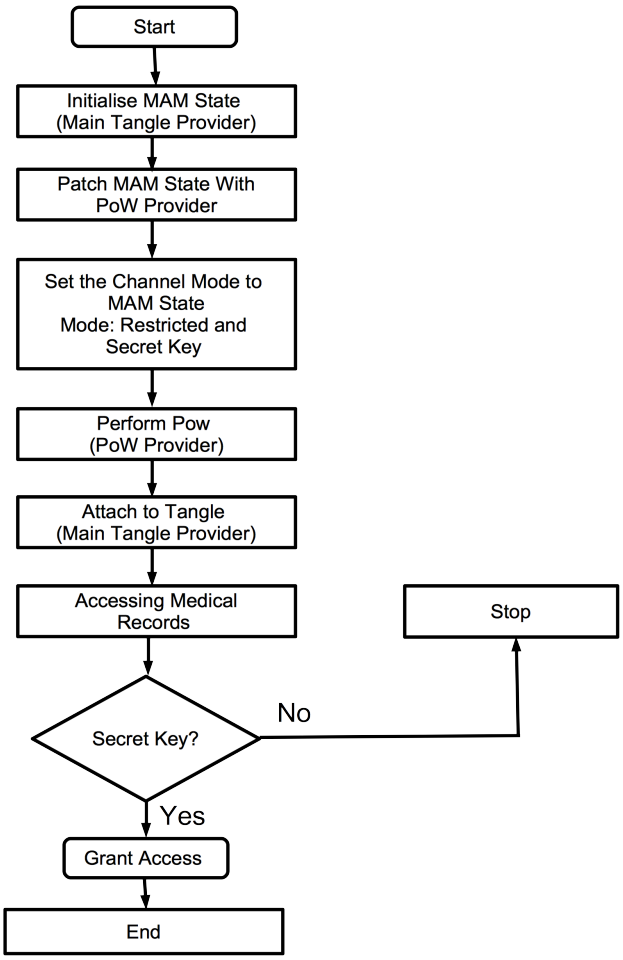

Fig. 6.5 PADI Approach

Figure 6.6 presents an illustration of how to set the privacy mode to restricted by selecting a secret key and connecting to a Devnet node. It also necessitates the use of a MAM URL explorer to view the interface. The process starts with the initialization of the MAM state. A function is required to convert ASCII data into trytes and store it in JSON format, prior to sending it to a MAM channel in restricted mode.

Figure 6.7 describes the rule for publishing data to the IOTA Tangle using the restricted mode. It requires the creation of a function that publishes MAM messages. A transaction data can be of any type e.g., temperature, humidity etc. This depends on the IoT application being developed.

```
(async function () {
      const mode = 'restricted'
      const secretKey = 'VERYSECRETKEY'
      const wrongsecretKey = 'WRONGSECRETKEY'
      const provider = 'https://nodes.devnet.iota.org'
      // Initialise the MAM State
        let mamState = Mam.init(provider)
        mamState = Mam.changeMode(mamState, mode, secretKey)
        // Publish to the tangle
        const publish = async packet => {
            // Create a MAM Payload - STRING OF TRYTES outputHtml.
                ↪ innerHTML += `Published:

            const trytes = asciiToTrytes(JSON.stringify(packet))
            const message = Mam.create(mamState, trytes)

            // Save new mamState
            mamState = message.state

            // Attach the payload
            await Mam.attach(message.payload, message.address, 3, 9)

            outputHtml.innerHTML += `Published: ${packet}<br/>`;
            return message.root
        }
```

Fig. 6.6 A Restricted Mode Example

```
const publishAll = async () => {

   outputHtml.innerHTML += '<h1>Publishing Data to Tangle using MAM
      ↪ in Restricted Mode </h1> <h3> Sending Data From IoT Devices
      ↪ </h3>';

     const root = await publish(' Mote 1 Data')
     await publish('Mote 2 Data')
     await publish('Mote 3 Data')

           return root
        }
```

Fig. 6.7 Publishing Data using Restricted Mode

6.5.1 PADI: A Healthcare Application Scenario

In the healthcare industry, IoT devices gather health related data such as blood pressure readings, temperature and heart rate readings (Zhang et al., 2018). DLT has a crucial role in the healthcare industry as it provides valuable features and solutions such as privacy, security, Transparency and decentralization, which has the potential to address significant challenges in healthcare systems (McGhin et al., 2019). The performance and efficiency of healthcare systems hinge on the ability of different software applications and technology platforms to securely communicate, exchange transaction data, and effectively access the data that is exchanged across healthcare organisations, which is facilitated by interoperability.

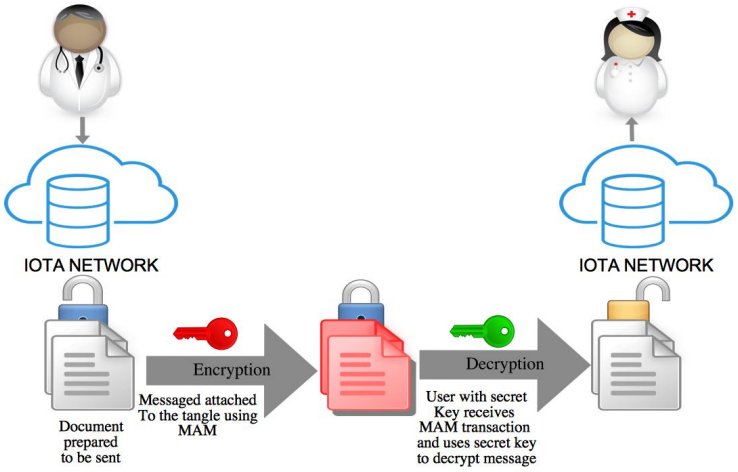

Fig. 6.8 PADI Healthcare Application

Effective data access management is crucial in healthcare applications, and thus the IOTA network utilizes MAM to encrypt transactions, ensuring that only authorized users can access the data when a transaction is recorded.

By storing data on the IOTA Tangle, nurses, doctors and patients can control who has access to specific parts of the data. When a user requests access to a patient's personal information, the algorithm checks their credentials and either grants or denies access based on the requesting user's rights to view the data. It's important to note that access to health records is secured by a secret key, preventing unauthorized individuals from viewing personal information. To summarize, the use of IOTA technology, especially MAM, to restrict data access guarantees that only authorized users will have access to data when necessary and that confidential information is only visible to those who are authorized. The access to healthcare records is based on a secret key, which keeps private information confidential from unauthorized users of the system.

6.6 Experimental Analysis

In this section, the experimental findings of the proposed PADI approach is reported, which indicates its effectiveness in attaining privacy and security for IoT applications.

6.6.1 Environment Setup

The PADI approach presented in this thesis is based on Node JS and utilizes the iota.lib.js library (IOTA, 2018) to implement transaction-related features, such as addresses, multi-signatures, and broadcasting. The iota.lib.js library contains the instructions for executing and retrieving transactions. A network simulation was created using the IOTA devnet to show how clients communicate with an IOTA full node 2 to send and receive data using MAM. Additionally, another IOTA full node was installed on a private local network for the purpose of performing Proof-of-Work (PoW).

The PADI approach centers around utilizing the public and restricted modes of IOTA's Masked Authenticated Messaging (MAM) protocol. By using public mode, the accuracy and consistency of the data can be guaranteed, while in restricted mode, a secret key is necessary to access the data. This is particularly beneficial in the healthcare sector where authorization is necessary to access sensitive information.

6.6.2 Results

The simulation of data transmission in this thesis involved sending transactions to the IOTA Tangle using public mode. The result of the retrieved transaction data revealed that a subscribing user only needs a root, which acts as the encryption and decryption key. Transactions that are sent using public mode guarantee the authenticity and confidentiality of the data by confirming its origin. The public mode takes the root and applies it as the address of the MAM transaction (channel ID), which enables users to locate and decrypt messages through public mode.

Authorised Access: The restricted mode in MAM provides a level of privacy for the data stored on the IOTA Tangle by granting specific access control to it. Only those with the proper secret key are able to decode the transaction data. To access the information, the user must first receive permission and then retrieve the transaction data, which can be decrypted using the secret key.

Unauthorised Access The author also tested the ability of the PADI approach to deny access when a wrong secret key is entered. Attempts were made to access data stored in the tangle using incorrect secret keys which were unauthorised (see Figure 6.9).

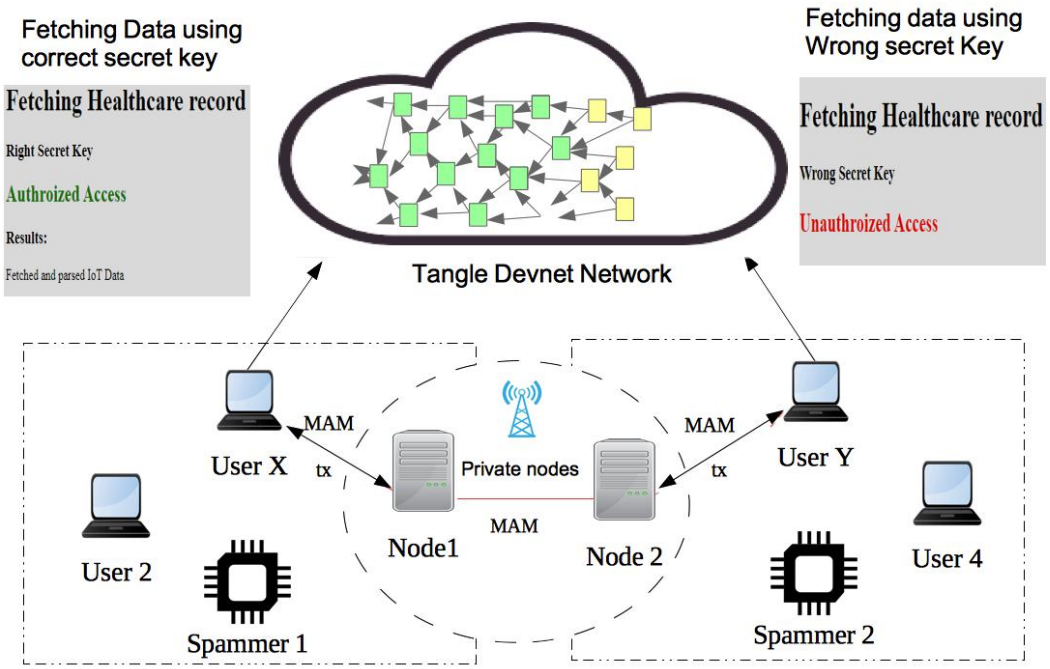

Fig. 6.9 Access Control Using the Tangle

6.6.3 Discussion

In this chapter, the formulation of a secure and private approach to missing data recovery using the IOTA Tangle is presented, which previous research has not accounted for. By exploiting the functionalities of the IOTA Tangle, missing data can be recovered across distributed sites without sharing subject level information, thereby preserving the content of the information held within sites in a DHDN. An Average Site Mixture (AvSM) model is developed in this thesis, which recovers missing data by using parameter estimates from a publishing site transmitted using MAM. Overall, this approach shows promising results when compared to benchmark techniques and is useful for time series imputation in real-world settings.

Security is a fundamental requirement for most IoT applications, which makes IOTA and enabling technology for various IoT applications. This chapter also presents a Privacy Aware Distributed Intelligence Framework (PADI) using IOTA's MAM protocol, which provides solutions to various IoT concerns as described below;

Confidentiality: Messages and transactions are only accessible to intended recipients and the Tangle provides confidentiality in such cases by encrypting data. Every transaction that is stored in the Tangle or communicated between nodes is encrypted and the MAM protocol adds an extra layer of encryption to transactions executed over the Tangle. Consequently, it enables only the authorised users to communicate with the data. Therefore, data confidentiality is well sustained when using the Tangle.

Authorisation: This is another important factor in industrial applications as it enables granular access to data. Specific users will have access to important data and can control the number of users that can access such data. In the context of the proposed PADI approach, only the user with direct access to healthcare data can share information with other stakeholders.

Integrity: Transmitted messages are tamper-proof and an attacker will not be able to change the data. The Tangle offers this feature, as the ledger has built-in integrity which is

immutable. The information that has been stored in the tangle cannot be changed or deleted. Therefore, integrity is preserved when using this type of DLT.

Availability: This is a fundamental security condition and this requirement is achieved in the IOTA Tangle. When nodes are up and running, transactions can easily be sent with convenience. However, if the internet connection goes down, the Tangle has another important feature, which is the offline capability, where nodes can still issue a transaction, while offline, but when the internet connection comes back, the re-attachment of all transactions to the main Tangle will be required. In addition, it has another significant feature, which is decentralisation. This removes the single point of failure from the system. Although this is not addressed in the proposed PADI framework, it presents potentials for future research.

Chapter 7

Conclusions and Future Work

7.1 Conclusions

Missing values are frequently observed in small to large data sets, requiring researchers to develop intelligent approaches for their recovery. Specialists sometimes ignore cases with missing values in their analysis, which often lead to biased conclusions. A wide range of techniques for handling missing values are available for static data cases, therefore it is important to develop more intelligent imputation techniques and incorporate new criteria for choosing optimal recovery solutions for real-life cases where there is no ground truth. As more sensitive data are being generated and transmitted over IoT networks today, research has also shown the need for adapting more secure and private solutions in missing data recovery. In conclusion, this thesis investigated the problem of missing data and its impact on downstream learning tasks with regression and classification problems. Further investigations were carried out into the issue of privacy protection during missing data recovery.

The author presented novel algorithms tailored to different use cases and demonstrated their viability against benchmark techniques used in sensor networks and clinical studies. The author further investigated the issue of privacy in missing data scenarios where sensitive data is concerned and developed a privacy-preserving approach for missing data recovery

in distributed systems. In this concluding chapter, a description of the relationship between previous chapters is presented and the main conclusions from this study as a whole is highlighted.

In the first part of this thesis, the theoretical underpinnings of missing data and privacy concerns in data mining processes were presented. It is observed that in order to develop an efficient strategy for handling missing data, it is important to understand the patterns and mechanisms in which missing data occurs in a data set. A clear implication of some of these missing data mechanisms can be seen from the experiments conducted in Chapter 4 and 5. The author leveraged this observation and laid the foundation for the initial experiments by reviewing a range of statistical and machine learning approaches to treating missing data. In addition, a review of relevant theories relating to privacy was conducted which served as a foundation for the solution proposed in Chapter 6 of this thesis.

In the second part of this thesis, the author developed the $k-BFMVI$ algorithm and evaluated its performance on downstream regression tasks for the calibration of on field air quality sensors whilst considering the different missing data mechanisms identified in the first part of this thesis. The proposed $k-BFMVI$ algorithm showed the overall best imputation performance when compared to other benchmark methods. An overall performance gain was observed in the learning algorithms when trained on the data set generated from the proposed algorithm. The proposed imputation algorithm showed a trade-off between imputation accuracy and complexity due to the performance cost of the sub-techniques embedded in the algorithm. The $med.BFMVI$ technique was further proposed as an extension of the $k-BFMVI$ algorithm. This algorithm was tailored to real-world clinical applications with downstream classification problems. A lower view of this algorithm was presented, showing the performance of each sub-technique benchmarked with other state-of-the-art imputation approaches. Overall, the $med.BFMVI$ algorithm showed promising imputation performance leading to an overall best accuracy when trained on classification algorithms. Overall, the

proposed algorithms presented good results which could be useful for real-life data mining processes.

In the third part of this thesis, the author takes knowledge from previous research and developed a privacy preserving framework for missing data recovery in distributed systems. The simulations conducted show the usefulness of the IOTA Tangle and MAM in ensuring a more secure and private approach to missing data recovery for cases where access to relevant covariates is limited for the purpose of preserving privacy, such as in clinical data sets. An Average Site Mixture (AvSM) approach was proposed for imputing missing values using parameter estimates from both sites without the need for sharing subject level information across each site.

Although, values imputed in missing instances may behave well and show consistency with other attribute values, some imputation procedures may be potentially harmful due to the fact that imputation algorithms are only able to approximate the actual values that are missing. Overall, the contributions of this thesis are complimentary and will have huge impacts in real-world application scenarios.

7.2 Future work

The work produced in this thesis contributes towards downstream and privacy preserving missing data recovery for IoT systems with static data. It also presents an opening for further research in this field. Whilst the contributions presented in this thesis can be adapted to various fields, there remain open areas for further investigations.

- The patterns of missing data considered in this thesis only reflected the missing completely at random (MCAR) and missing at random (MAR) assumptions. Further investigations will be required to determine the extent that these assumptions affect different missing data recovery techniques. In other words, where a different missing

data mechanism is concerned, such as not missing at random (NMAR) mechanism or structural missing data assumption identified in section 5.2.5, will the proposed approach present any performance gain?

- In order to increase the strength of new missing data imputation algorithms, the proposed error score function presented in Chapter 4 can be improved to provide a more robust selection criteria to enable sound judgement on the choice of imputation techniques applied in real-life missing data cases.

- Chapter 6 of this thesis introduces a privacy-preserving approach to missing data recovery using IOTA's MAM protocol and develops the AvSM technique by exploiting the functionalities of the Tangle. Although the AvSM approach outperforms other fast imputation methods identified and preserves privacy by imputing missing values without the need for sharing subject level data, there remains a problem that requires further investigation. The simulation results show that the proposed AvSM performs well in stochastic conditions where there is no uniformity in data imputes over time. Future research is however required to assess the viability of imputing over the Tangle when different data distributions are concerned. E.g. Linear distributions.

- As large numbers of sensor data are generated and transmitted over the IOTA Tangle, more explorations could be conducted to further exploit the IOTA Tangle for efficient privacy preserving missing data recovery by allocating a designated node to perform missing data calculations before transmitting to subscribing sites, similar to PoW computations carried out by designated nodes.

- The problem statements considered in this thesis focused on static data. More work is required to extend the proposed missing data algorithms for cases with dynamic data.

References

Abdullah, S., Arshad, J., Khan, M. M., Alazab, M., and Salah, K. (2022). Prised tangle: a privacy-aware framework for smart healthcare data sharing using iota tangle. *Complex & Intelligent Systems*, pages 1–19.

Agbo, B., Al-Aqrabi, H., Hill, R., and Alsboui, T. (2022). Missing data imputation in the internet of things sensor networks. *Future Internet*, 14(5):143.

Agbo, B., Qin, Y., and Hill, R. (2020). Best fit missing value imputation (bfmvi) algorithm for incomplete data in the internet of things. In *IoTBDS*, pages 130–137.

Agrawal, R. and Srikant, R. (2000). Privacy-preserving data mining. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pages 439–450.

Akaike, H. (1974). A new look at the statistical model identification. *IEEE transactions on automatic control*, 19(6):716–723.

Al-Aqrabi, H., Johnson, A. P., Hill, R., Lane, P., and Alsboui, T. (2020). Hardware-intrinsic multi-layer security: a new frontier for 5g enabled iiot. *Sensors*, 20(7):1963.

Allison, P. D. (2000). Multiple imputation for missing data: A cautionary tale. *Sociological methods & research*, 28(3):301–309.

Allmer, T. (2011). A critical contribution to theoretical foundations of privacy studies. *Journal of Information, Communication and Ethics in Society*.

Almas, A. (2018). *Depression and Cardiovascular Diseases*. PhD thesis, Karolinska Institutet (Sweden).

Alpaydin, E. (2020). *Introduction to machine learning*. MIT press.

Alsaber, A. R., Pan, J., and Al-Hurban, A. (2021). Handling complex missing data using random forest approach for an air quality monitoring dataset: a case study of kuwait environmental data (2012 to 2018). *International Journal of Environmental Research and Public Health*, 18(3):1333.

Amirteimoori, A. and Kordrostami, S. (2010). A euclidean distance-based measure of efficiency in data envelopment analysis. *Optimization*, 59(7):985–996.

Anderson, T. W. (1957). Maximum likelihood estimates for a multivariate normal distribution when some observations are missing. *Journal of the american Statistical Association*, 52(278):200–203.

Back, A. (2002). Hashcash-a denial of service counter-measure.

Baraldi, A. N. and Enders, C. K. (2010). An introduction to modern missing data analyses. *Journal of school psychology*, 48(1):5–37.

Barnaghi, P., Bermudez-Edo, M., and Tönjes, R. (2015). Challenges for quality of data in smart cities. *Journal of Data and Information Quality (JDIQ)*, 6(2-3):1–4.

Bartolomeu, P. C., Vieira, E., and Ferreira, J. (2018). Iota feasibility and perspectives for enabling vehicular applications. In *2018 IEEE Globecom Workshops (GC Wkshps)*, pages 1–7. IEEE.

Bashir, F. (2019). *Handling of Missing Values in Static and Dynamic Data Sets*. PhD thesis, University of Sheffield.

Bilmes, J. A. (1998). A gentle tutorial of the em algorithm and its application to parameter estimation for gaussian mixture and hidden markov models. *International Computer Science Institute*, 4(510):126.

Blackwell, M., Honaker, J., and King, G. (2017). A unified approach to measurement error and missing data: overview and applications. *Sociological Methods & Research*, 46(3):303–341.

Breiman, L. (2001). Random forests. *Machine Learning*, 45:5–32.

Breiman, L., Friedman, J., Olshen, R., and Stone, C. (1984). Classification and regression t rees (monterey, california: Wadsworth).

Burpee, D., Dabaghi, H., Jackson, L., Kwamena, F., Richter, J., Rusnov, T., Friedman, K., Mansueti, L., and Meyer, D. (2006). Us-canada power system outage task force: final report on the implementation of task force recommendations.

Byabazaire, J., O'Hare, G., and Delaney, D. (2020). Data quality and trust: Review of challenges and opportunities for data sharing in iot. *Electronics*, 9(12):2083.

Cachin, C., Guerraoui, R., and Rodrigues, L. (2011). *Introduction to reliable and secure distributed programming*. Springer Science & Business Media.

Callahan, A. and Shah, N. H. (2017). Machine learning in healthcare. In *Key Advances in Clinical Informatics*, pages 279–291. Elsevier.

Caruana, E. J., Roman, M., Hernández-Sánchez, J., and Solli, P. (2015). Longitudinal studies. *Journal of Thoracic Disease*, 7(11).

Castro, M., Liskov, B., et al. (1999). Practical byzantine fault tolerance. In *OSDI*, volume 99, pages 173–186.

Chang, C., Deng, Y., Jiang, X., and Long, Q. (2020). Multiple imputation for analysis of incomplete data in distributed health data networks. *Nature communications*, 11(1):1–11.

Chechik, G., Heitz, G., Elidan, G., Abbeel, P., and Koller, D. (2006). Max-margin classification of incomplete data. *Advances in Neural Information Processing Systems*, 19.

Chen, Y., Lv, Y., and Wang, F.-Y. (2019). Traffic flow imputation using parallel data and generative adversarial networks. *IEEE Transactions on Intelligent Transportation Systems*, 21(4):1624–1630.

Cutler, A., Cutler, D. R., and Stevens, J. R. (2012). Random forests. In *Ensemble machine learning*, pages 157–175. Springer.

Danzi, P., Kalør, A. E., Stefanović, Č., and Popovski, P. (2019). Delay and communication tradeoffs for blockchain systems with lightweight iot clients. *IEEE Internet of Things Journal*, 6(2):2354–2365.

De Vito, S., Massera, E., Piga, M., Martinotto, L., and Di Francia, G. (2008). On field calibration of an electronic nose for benzene estimation in an urban pollution monitoring scenario. *Sensors and Actuators B: Chemical*, 129(2):750–757.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22.

Deng, Y., Jiang, X., and Long, Q. (2020). Privacy-preserving methods for vertically partitioned incomplete data. In *AMIA Annual Symposium Proceedings*, volume 2020, page 348. American Medical Informatics Association.

Duda, R. O., Hart, P. E., and Stork, D. G. (2012). Pattern classification. 605 third avenue.

Ehrlinger, L., Grubinger, T., Varga, B., Pichler, M., Natschläger, T., and Zeindl, J. (2018). Treating missing data in industrial data analytics. In *2018 Thirteenth International Conference on Digital Information Management (ICDIM)*, pages 148–155. IEEE.

Elgesem, D. (1999). The structure of rights in directive 95/46/ec onthe protection of individuals with regard to theprocessing of personal data and the free movement ofsuch data. 1(4).

Emmanuel, T., Maupong, T., Mpoeleng, D., Semong, T., Banyatsang, M., and Tabona, O. (2021). A survey on missing data in machine learning.

Enders, C. K. (2006). A primer on the use of modern missing-data methods in psychosomatic medicine research. *Psychosomatic medicine*, 68(3):427–436.

Enders, C. K. (2010). *Applied missing data analysis*. Guilford press.

Farhangfar, A., Kurgan, L., and Dy, J. (2008). Impact of imputation of missing values on classification error for discrete data. *Pattern Recognition*, 41(12):3692–3705.

Fekade, B., Maksymyuk, T., Kyryk, M., and Jo, M. (2017). Probabilistic recovery of incomplete sensed data in iot. *IEEE Internet of Things Journal*, 5(4):2282–2292.

Frank, A. (2010). Uci machine learning repository. *http://archive. ics. uci. edu/ml*.

Gavison, R. (1980). Privacy and the limits of law. *The Yale law journal*, 89(3):421–471.

Genes, C. (2018). *Novel Matrix Completion Methods for Missing Data Recovery in Urban Systems*. PhD thesis, University of Sheffield.

Genes, C., Esnaola, I., Perlaza, S. M., Ochoa, L. F., and Coca, D. (2016). Recovering missing data via matrix completion in electricity distribution systems. In *2016 IEEE 17th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, pages 1–6. IEEE.

Gourieroux, C. and Monfort, A. (1981). On the problem of missing data in linear models. *The Review of Economic Studies*, 48(4):579–586.

Graham, J. W. (2009). Missing data analysis: Making it work in the real world. *Annual review of psychology*, 60:549–576.

Handy, P. (2017). Introducing masked authenticated messaging. *IOTA: Berlin, Germany*.

Hastie, T., Tibshirani, R., Friedman, J., and Franklin, J. (2004). The elements of statistical learning: Data mining, inference, and prediction. *Math. Intell.*, 27:83–85.

Izonin, I., Kryvinska, N., Tkachenko, R., and Zub, K. (2019). An approach towards missing data recovery within iot smart system. *Procedia Computer Science*, 155:11–18.

Jagannathan, G. and Wright, R. N. (2008). Privacy-preserving imputation of missing data. *Data & Knowledge Engineering*, 65(1):40–56.

Javed, A. R., Fahad, L. G., Farhan, A. A., Abbas, S., Srivastava, G., Parizi, R. M., and Khan, M. S. (2021). Automated cognitive health assessment in smart homes using machine learning. *Sustainable Cities and Society*, 65:102572.

Karkouch, A., Mousannif, H., Al Moatassime, H., and Noel, T. (2016). Data quality in internet of things: A state-of-the-art survey. *Journal of Network and Computer Applications*, 73:57–81.

Kotsiantis, S., Kostoulas, A., Lykoudis, S., Argiriou, A., and Menagias, K. (2006). Filling missing temperature values in weather data banks. In *2006 2nd IET International Conference on Intelligent Environments-IE 06*, volume 1, pages 327–334. IET.

Lai, L. L., Zhang, H. T., Lai, C. S., Xu, F. Y., and Mishra, S. (2013). Investigation on july 2012 indian blackout. In *2013 International Conference on Machine Learning and Cybernetics*, volume 1, pages 92–97. IEEE.

Lakshminarayan, K., Harp, S. A., and Samad, T. (1999). Imputation of missing data in industrial databases. *Applied intelligence*, 11(3):259–275.

Lamtzidis, O. and Gialelis, J. (2018). An iota based distributed sensor node system. In *2018 IEEE Globecom Workshops (GC Wkshps)*, pages 1–6. IEEE.

Lee, G. H., Han, J., and Choi, J. K. (2021). Mpdist-based missing data imputation for supporting big data analyses in iot-based applications. *Future Generation Computer Systems*, 125:421–432.

Lee, M., An, J., and Lee, Y. (2019). Missing-value imputation of continuous missing based on deep imputation network using correlations among multiple iot data streams in a smart space. *IEICE TRANSACTIONS on Information and Systems*, 102(2):289–298.

Lee, W.-C., Hanson, B. A., and Brennan, R. L. (2002). Estimating consistency and accuracy indices for multiple classifications. *Applied Psychological Measurement*, 26(4):412–432.

Liang, J. and Bentler, P. M. (2004). An em algorithm for fitting two-level structural equation models. *Psychometrika*, 69(1):101–122.

Little, R. J. (1992). Regression with missing x's: a review. *Journal of the American statistical association*, 87(420):1227–1237.

Little, R. J. and Rubin, D. B. (1987). *Statistical analysis with missing data*, volume 793. John Wiley & Sons.

Little, R. J. and Rubin, D. B. (2019). *Statistical analysis with missing data*, volume 793. John Wiley & Sons.

Loy-Benitez, J., Heo, S., and Yoo, C. (2020). Imputing missing indoor air quality data via variational convolutional autoencoders: Implications for ventilation management of subway metro systems. *Building and Environment*, 182:107135.

Luengo, J., García, S., and Herrera, F. (2012). On the choice of the best imputation methods for missing values considering three groups of classification methods. *Knowledge and information systems*, 32(1):77–108.

Maillo, J., Ramírez, S., Triguero, I., and Herrera, F. (2017). knn-is: An iterative spark-based design of the k-nearest neighbors classifier for big data. *Knowledge-Based Systems*, 117:3–15.

Manogaran, G., Rawal, B. S., Saravanan, V., Kumar, P. M., Martínez, O. S., Crespo, R. G., Montenegro-Marin, C. E., and Krishnamoorthy, S. (2020). Blockchain based integrated security measure for reliable service delegation in 6g communication environment. *Computer Communications*, 161:248–256.

Marlin, B. (2008). *Missing data problems in machine learning*. PhD thesis.

Marshall, A., Altman, D. G., and Holder, R. L. (2010). Comparison of imputation methods for handling missing covariate data when fitting a cox proportional hazards model: a resampling study. *BMC medical research methodology*, 10(1):1–10.

Martin, K. (2012). Information technology and privacy: conceptual muddles or privacy vacuums? *Ethics and Information Technology*, 14(4):267–284.

Mazzeo, N. A. and Venegas, L. E. (2005). Evaluation of turbulence from traffic using experimental data obtained in a street canyon. *International journal of environment and pollution*, 25(1-4):164–176.

McGhin, T., Choo, K.-K. R., Liu, C. Z., and He, D. (2019). Blockchain in healthcare applications: Research challenges and opportunities. *Journal of Network and Computer Applications*, 135:62–75.

Mehrwald, P., Treffers, T., Titze, M., and Welpe, I. (2019). Blockchain technology application in the sharing economy: a proposed model of effects on trust and intermediation.

Mohamed, C., Sedory, S. A., and Singh, S. (2018). Improved mean methods of imputation. *Statistics, Optimization & Information Computing*, 6(4):526–535.

Murdoch, T. B. and Detsky, A. S. (2013). The inevitable application of big data to health care. *Jama*, 309(13):1351–1352.

Muthén, B. and Shedden, K. (1999). Finite mixture modeling with mixture outcomes using the em algorithm. *Biometrics*, 55(2):463–469.

Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system. *Decentralized Business Review*, page 21260.

Neal, R. M. and Hinton, G. E. (1998). A view of the em algorithm that justifies incremental, sparse, and other variants. In *Learning in graphical models*, pages 355–368. Springer.

Neale, M. C., Boker, S. M., Xie, G., and Maes, H. M. (1999). Statistical modeling. *Richmond, VA: Department of Psychiatry, Virginia Commonwealth University*.

Nelder, J. A. and Wedderburn, R. W. (1972). Generalized linear models. *Journal of the Royal Statistical Society: Series A (General)*, 135(3):370–384.

Obermeyer, Z. and Emanuel, E. J. (2016). Predicting the future—big data, machine learning, and clinical medicine. *The New England journal of medicine*, 375(13):1216.

Okafor, N. (2021). Missing data imputation on iot sensor networks: Implications for on-site sensor calibration.

Osman, M. S., Abu-Mahfouz, A. M., and Page, P. R. (2018). A survey on data imputation techniques: Water distribution system as a use case. *IEEE Access*, 6:63279–63291.

Peugh, J. L. and Enders, C. K. (2004). Missing data in educational research: A review of reporting practices and suggestions for improvement. *Review of educational research*, 74(4):525–556.

Popov, S. (2018). The tangle. *White paper*, 1(3).

Powell, H. L. (2012). *Estimating air pollution and its relationship with human health*. PhD thesis, University of Glasgow.

Raja, P. and Thangavel, K. (2020). Missing value imputation using unsupervised machine learning techniques. *Soft Computing*, 24(6):4361–4392.

Raudenbush, S. W. and Bryk, A. S. (2002). *Hierarchical linear models: Applications and data analysis methods*, volume 1. sage.

Rubin, D. B. (1976). Inference and missing data. *Biometrika*, 63(3):581–592.

Rubin, D. B. (1984). Bayesianly justifiable and relevant frequency calculations for the applied statistician. *The Annals of Statistics*, pages 1151–1172.

Rubin, D. B. (1996). Multiple imputation after 18+ years. *Journal of the American statistical Association*, 91(434):473–489.

Sagirlar, G. (2018). *Enhancing data privacy and security in Internet of Things through decentralized models and services*. PhD thesis, Università degli Studi dell'Insubria.

Sanjar, K., Bekhzod, O., Kim, J., Paul, A., and Kim, J. (2020). Missing data imputation for geolocation-based price prediction using knn–mcf method. *ISPRS International Journal of Geo-Information*, 9(4):227.

Schafer, J. L. (1997). *Analysis of incomplete multivariate data*. CRC press.

Schafer, J. L. and Graham, J. W. (2002). Missing data: our view of the state of the art. *Psychological methods*, 7(2):147.

Schwarz, G. (1978). Estimating the dimension of a model. *The annals of statistics*, pages 461–464.

Sharpe, P. K. and Solly, R. (1995). Dealing with missing values in neural network-based diagnostic systems. *Neural Computing & Applications*, 3(2):73–77.

Shickel, B., Tighe, P. J., Bihorac, A., and Rashidi, P. (2017). Deep ehr: a survey of recent advances in deep learning techniques for electronic health record (ehr) analysis. *IEEE journal of biomedical and health informatics*, 22(5):1589–1604.

Song, Q. and Shepperd, M. (2007). Missing data imputation techniques. *International journal of business intelligence and data mining*, 2(3):261–291.

Stekhoven, D. J. and Bühlmann, P. (2012). Missforest—non-parametric missing value imputation for mixed-type data. *Bioinformatics*, 28(1):112–118.

Stolpe, M. (2016). The internet of things: Opportunities and challenges for distributed data analysis. *Acm Sigkdd Explorations Newsletter*, 18(1):15–34.

Suvarna, M., Büth, L., Hejny, J., Mennenga, M., Li, J., Ng, Y. T., Herrmann, C., and Wang, X. (2020). Smart manufacturing for smart cities—overview, insights, and future directions. *Advanced Intelligent Systems*, 2(10):2000043.

Tavani, H. T. and Moor, J. H. (2001). Privacy protection, control of information, and privacy-enhancing technologies. *ACM Sigcas Computers and Society*, 31(1):6–11.

Tresp, V., Ahmad, S., and Neuneier, R. (1993). Training neural networks with deficient data. *Advances in neural information processing systems*, 6.

Truong, N. B., Sun, K., and Guo, Y. (2019). Blockchain-based personal data management: from fiction to solution. In *2019 IEEE 18th International Symposium on Network Computing and Applications (NCA)*, pages 1–8. IEEE.

UCI (Accessed: Feb 2, 2022). https://archive.ics.uci.edu/ml/datasets/air+quality.

Uddin, M. A., Stranieri, A., Gondal, I., and Balasubramanian, V. (2021). A survey on the adoption of blockchain in iot: Challenges and solutions. *Blockchain: Research and Applications*, page 100006.

UN (1948). Universal declaration of human rights.

Van Ginkel, J. R., Van der Ark, L. A., Sijtsma, K., and Vermunt, J. K. (2007). Two-way imputation: A bayesian method for estimating missing scores in tests and questionnaires, and an accurate approximation. *Computational Statistics & Data Analysis*, 51(8):4013–4027.

Vink, G. (2022). Roderick j. little and donald b. rubin: Statistical analysis with missing data.

Wahlstrom, K. and Fairweather, N. B. (2013). Privacy, the theory of communicative action and technology.

Wand, M. P. (2000). A comparison of regression spline smoothing procedures. *Computational Statistics*, 15(4):443–462.

Wang, W., Hoang, D. T., Hu, P., Xiong, Z., Niyato, D., Wang, P., Wen, Y., and Kim, D. I. (2019). A survey on consensus mechanisms and mining strategy management in blockchain networks. *IEEE Access*, 7:22328–22370.

Ware, J. H., Harrington, D., Hunter, D. J., and D'Agostino Sr, R. B. (2012). Missing data.

Westin, A. F. (1970). Privacy and freedom london. *The Bodley Head*.

Wilks, S. S. (1932). Moments and distributions of estimates of population parameters from fragmentary samples. *The Annals of Mathematical Statistics*, 3(3):163–195.

Wood, A. M., White, I. R., and Thompson, S. G. (2004). Are missing outcome data adequately handled? a review of published randomized controlled trials in major medical journals. *Clinical trials*, 1(4):368–376.

Zhang, P., Schmidt, D. C., White, J., and Lenz, G. (2018). Chapter one - blockchain technology use cases in healthcare. In Raj, P. and Deka, G. C., editors, *Blockchain Technology: Platforms, Tools and Use Cases*, volume 111 of *Advances in Computers*, pages 1–41. Elsevier.

Zhang, Q., Yang, L. T., Chen, Z., and Xia, F. (2015). A high-order possibilistic c-means algorithm for clustering incomplete multimedia data. *IEEE Systems Journal*, 11(4):2160–2169.

Zhao, L., Chen, Z., Yang, Z., Hu, Y., and Obaidat, M. S. (2018). Local similarity imputation based on fast clustering for incomplete data in cyber-physical systems. *IEEE Systems Journal*, 12(2):1610–1620.