Combined Supervised and Unsupervised Learning to Identify Subclasses of Disease for Better Prediction

A thesis submitted for the degree of Doctor of Philosophy by

Awad Alsaid Alyousef

Department of computer science

Brunel University

Supervisor: Dr Allan Tucker

October 2022

Abstract

Disease subtyping, which aids in the development of personalised treatments, remains a challenge in data analysis because of the many different ways to group patients based upon their data. However, if I can identify subclasses of disease, this will help to develop better models that are more specific to individuals and should therefore improve prediction and understanding of the underlying characteristics of the disease in question. In addition, patients might suffer from multiple disease complications. Models that are tailored to individuals could improve both prediction of multiple complications and understanding of underlying disease characteristics. However, AI models can become outdated over time due to either sudden changes in the underlying data, such as those caused by new measurement methods, or incremental changes, such as the ageing of the study population. This thesis proposes a new algorithm that integrates consensus clustering methods with classification in order to overcome issues with sample bias. The method was tested on a freely available dataset of real-world breast cancer cases and data from a London hospital on systemic sclerosis, a rare and potentially fatal condition. The results show that nearest consensus clustering classification improves accuracy and prediction significantly when this algorithm is compared with competitive similar methods. In addition, this thesis proposes a new algorithm that integrates latent class models with classification. The new algorithm uses latent class models to cluster patients within groups; this results in improved classification and aids in the understanding of the underlying differences of the discovered groups. The method was tested on data from patients with systemic sclerosis (SSc), a rare and potentially fatal condition, and coronary heart disease. Results show that the latent class multi-label classification (MLC) model improves accuracy when compared with competitive similar methods. Finally, this thesis implemented the updated concept drift method (DDM) to monitor AI models over time and detect drifts when they occur. The method was tested on data from patients with SSc and patients with coronavirus disease (COVID).

Keywords: Classification; Consensus clustering; Disease subgroup discovery, Latent Class Analysis, Multi-Label Classification, Concept Drift.

Acknowledgements

Without a doubt, finishing my thesis was one of the most challenging and achievement projects I have ever worked on. I would therefore like to thank Dr. Allan Tucker for all of his help and support throughout this journey. This thesis would not have been possible without his insightful recommendations and invaluable assistance. I would also like to thank the Royal London Hospital as well as the Medicines and Healthcare products Regulatory Agency (MHRA) for assistance in gaining access to the data.

I would like to express my appreciation to my mother, my wife, and my children for their assistance and their patience throughout this difficult time. I could not have written this thesis without their help and cheerful faces. Finally, I would like to dedicate this thesis to my father and all of my brothers who were killed in the dirty war in Syria. I need to say you left me alone, but I will never forget your smiles, kind words, and encouragement.

Publications

The following publications have resulted from the research presented in this thesis:

- Alyousef, A. A., Nihtyanova, S., Denton, C., Bosoni, P., Bellazzi, R., & Tucker, A. (2017). Nearest consensus clustering classification to identify subclasses and predict disease. Workshop on Advanced Predictive Models in Healthcare page 26-35.

- Alyousef, A. A., Nihtyanova, S., Denton, C., Bosoni, P., Bellazzi, R., & Tucker, A. (2018). Nearest consensus clustering classification to identify subclasses and predict disease. *Journal of Healthcare Informatics Research, 2*(23), 402-422. https://doi.org/10.1007/s41666-018-0029-6

- Alyousef, A. A., Nihtyanova, S., Denton, C., Bosoni, P., Bellazzi, R., & Tucker, A. (2019). *Latent class multi-label classification to identify subclasses of disease for improved prediction* [Poster]. 2019 IEEE 32nd International Symposium on Computer-Based Medical Systems, Cordoba, Spain. https://doi.org/1010.1109/CBMS.2019.00109

Contents

List of Figures

List of Tables

Abbreviations

AI	**Artificial Intelligence**
ML	**Machine Learning**
PM	**Personalized Medicine**
DT	**Decision Tree**
MLC	**Multi Label Classification**
NB	**Naïve Bayes**
BN	**Bayesian Networks**
ANN	**Artificial Neural Networks**
SSc	**Systemic Sclerosis**
COVID	**Coronavirus Disease**
RF	**Random Forests**
LCA	**Latent Class Analysis**
DDM	**Drift Detection Method**
BC	**Breast Cancer**
CC	**Consensus Clustering**
BR	**Binary Relevance**
LP	**Label Powerset**
MLNB	**Multi label classification Naïve Bayes**
LCNB	**Latent Class Analysis Naïve Bayes**
LCMLNB	**Latent Class Multi Label Naïve Bayes**
EDDM	**Early Drift Detection Method**
ADWIN	**Adaptive Windowing**
DDM-OCI	**Drift Detection Method Recall Metric**

DBSCAN	**Density-based spatial clustering of applications with noise**
ECG	**Electrocardiogram**
NHS	**National Health Service**
ED	**Emergency department**
ATA	**Anti-topoisomerase I antibodies**
ACA	**Anti-centromere antibodies**
ARA	**Anti-RNA polymerase antibodies**
PAH	**Pulmonary arterial hypertension**
PF	**Pulmonary fibrosis**
dcSSc	**Diffuse systemic sclerosis**
lcSSc	**Limited systemic sclerosis**
Hb	**Haemoglobin concentration**
Cr	**Creatinine**
FVC	**Forced vital capacity**
DLCO	**Carbon monoxide diffusing capacity**
T2RIP	**The number of months between the development of the sickness and mortality.**
T2PF	**The number of months between the commencement of the disease and the development of pulmonary fibrosis.**
T2PAH	**The number of months between the commencement of the disease and the onset of pulmonary arterial hypertension.**

ERR	**The error rate**
ACC	**The accuracy**
SN	**Sensitivity**
SP	**Specificity**
PPV	**Positive predictive value**
ROC	**Receiver operating characteristics**
c-means	**Fuzzy clustering algorithms**
ccRCC	**Clear cell renal cell carcinoma**
KNN	**K-nearest neighbours**
EM	**Expectation–maximization algorithm**
RNA	**Ribonucleic acid**
BIC	**Bayesian information criteria**
SVM	**Support vector machines**
LR	**Logistic Regression**

Chapter 1 Introduction

1.1 Overview

Early diagnosis of chronic diseases and the discovery of disease subclasses improve patient survival and reduce healthcare costs. Nowadays, chronic diseases, such as heart disease, diabetes, and cancer, have become a primary concern in the healthcare sector, which is struggling to find effective solutions. By 2025, the World Health Organisation predicts that 70% of all diseases will be chronic, meaning that they will require continuous care by healthcare providers. As a result, the prevalence of chronic diseases has increased both mortality and healthcare costs, which in certain countries may exceed economic capacity (Elton & O’Riordan, 2016).

Consequently, healthcare systems worldwide must treat patients more efficiently. For instance, the United States launched the Health Information Technology for Economic and Clinical Health Act in 2009 to help healthcare providers use electronic health records more efficiently (Blumenthal, 2010). In other words, the vast amount of medical data generated by healthcare providers and technologies are critical to improving patient care; these data can be used to tailor an appropriate treatment to each patient to achieve improvements in overall health (Vicente et al., 2020). It has become clear that incorporating medical data into healthcare systems delivers significant benefits in terms of accuracy, early diagnosis, identification of disease subclass, and potential to provide individualised patient diagnosis and treatment (Tucker et al., 2020).

Technologies that generate much medical data are critical because they allow healthcare professionals to better understand patients’ diseases and their progression, which increases overall care quality (Pastorino et al., 2019). Historically, physicians have been limited to using filing systems to access medical records to help with diagnosis and treatment assessment; however, they can accomplish this more efficiently by utilising computational methods to model diseases. Thus, physicians can use advanced computational methods to diagnose a patient’s disease and identify an appropriate, personalised treatment more accurately. Hence, artificial intelligence (AI) technologies, computer systems with the ability to solve tasks that normally require human intelligence, can be used to analyse and extract useful information from large medical datasets. Over the last several years, more and more AI methods have been developed, and they promise to revolutionise medicine. Researchers have suggested that modern AI technologies such as machine learning (ML) algorithms should be used in healthcare systems to create powerful predictive models (Handelman et al., 2018).

Because many healthcare challenges, such as making a diagnosis, classifying biological samples, and predicting health outcomes, can be improved by exploiting historical data, the healthcare industry is becoming increasingly reliant on advanced ML approaches. ML algorithms assist physicians in precisely analysing and diagnosing diseases to personalise patient treatment (Wiens & Wallace, 2016). As a result, ML techniques have been developed to turn massive amounts of data into knowledge to benefit the healthcare industry and improve patient care (Correa da Silva et al., 2020).

The future of ML in healthcare is bright as many companies are launching ML algorithms to accurately identify diseases at early stages. PM, based on individual health data, is another ML application that has received much research attention due to its ability to improve disease assessment and data collection. For instance, the SkinVision application allows patients to upload images of skin spots and thereby acquire personalised treatment plans for skin cancer (Kovalenko, 2020). Moreover, ML could prove valuable for rare disease subtyping by providing predictive models capable of identifying illness subclasses, which could help healthcare professionals to improve diagnosis and develop personalised treatments. ML could also play an essential role in predicting complications of multiple rare diseases by discovering unmeasured effects and thereby helping to build a robust learning model. However, AI models could become outdated over time due to sudden changes in the underlying data, such changes could be due to new measurement methods, or incremental changes, such as the aging of the study population.

1.2 Research motivation

Due to diagnostic challenges and a lack of knowledge, most rare diseases that affect a very small number of people (1 to 2,000 people) have no effective treatment. Worldwide, 350–400 million people suffer from rare diseases, most of which are caused by a hereditary gene mutation. Accurate diagnosis of rare diseases may take several years, significantly delaying treatment options; thus, patients with a rare disease may die before receiving a diagnosis. Furthermore, patients with the same rare disease may display different symptoms because a single disease could include subtypes, the identification of which has become increasingly important for improving personalised treatment. For example, the lungs may be affected in some systemic sclerosis (SSc) patients, while others may experience heart complications (Panopoulos et al., 2018). In brief, due to diagnostic delays, the lack of clinical expertise, insufficient information about rare diseases and these illnesses' life-threatening nature, advanced ML methods can play an important role in rare disease diagnosis and prognosis (Panopoulos et al., 2018).

In addition to disease subtypes, comorbidities, meaning the simultaneous presence of two or more diseases in a patient are one of the most pressing issues in healthcare. Comorbidities are classified as chronic conditions. For example, patients with type 2 diabetes are at high risk of developing both liver and bladder cancer. Multiple life-threatening comorbidities (complications) are associated with SSc, including arterial hypertension and depression. This means that patients with SSc, a rare disease, are likely to develop comorbid conditions that increase mortality. Hypokalaemic periodic paralysis, another rare genetic disorder, causes blood potassium levels to fall dangerously low, resulting in muscle weakness. Complications of this disease can affect the heart and the kidneys and cause breathing problems. It is difficult to diagnose hypokalaemic periodic paralysis due to lack of knowledge and its complicated symptoms; thus, many patients die before receiving an accurate diagnosis (Boban et al., 2019).

Rare disease diagnoses and the prediction of multiple comorbidities can be complicated by the presence of unmeasured risk factors, also known as hidden or latent variables. These hidden variables affect clinical trial outcomes. Detecting and explaining these hidden variables in medical datasets would improve the accuracy with which multiple complications can be predicted and to provide the provision of better PM. Understanding groups of patients based on newly discovered hidden variables, as well as better understanding disease progression and therefore providing more accurate diagnoses, would improve PM (Yousefi et al., 2018). As a result, an advanced ML approach should be created to construct a straightforward model by discovering hidden variables in rare disease medical data. Such an approach would enable the reliable prediction of multiple complications and the provision of better PM through the recognition of consensus groups of patients.

Predicting healthcare outcomes and complications over time is critical for providing better PM and improving healthcare management. These predictions would be based on electronic health records (patient data), meaning that the relationship between the patient's characteristics and their healthcare outcomes may change over time, affecting the ML model's performance unless the model is updated. This problem, called 'concept drift', occurs when the relationship between input data and the target variable changes over time (Gama et al., 2014). Methods for addressing this concept drift problem are urgently required.

Based on the aforementioned problems, this thesis focuses on ML methodologies for modelling SSc. It uses consensus advanced ML to identify disease subclasses, which will lead to better models that are more specific to individual groups of patients, and it builds a model that predict multiple complications by discovering hidden factors. Additionally, this thesis aims to implement a metric for detecting drift in ML models.

1.3 Research aims

The previous section presented motivations for using advanced ML algorithms to help healthcare providers classify and predict complications of rare diseases. This thesis aims to propose a new model that combines supervised ML with unsupervised ML to simultaneously identify subclasses and accurately predict SSc health outcomes. Additionally, it aims to build new models to predict multiple complications of SSc using a latent class variable to find robust groups of SSc patients. Furthermore, it aims to solve the concept drift problem by using a method for detecting a drift in ML models. Thus, these models will provide healthcare providers with a useful decision-making tool to support their choices. The contributions of this thesis are detailed in the following section.

1.4 Research contributions

The principal contributions of this research are as follows:

- **Exploration of state-of-the-art ML algorithms to model disease, with a focus on SSc:** Researchers have applied a set of ML algorithms, such as decision tree (DT), naïve Bayes (NB), k-means, random forest, and latent class analysis algorithms, to SSc datasets for clinical outcome predictions and patient clustering. These methods are analysed and evaluated using a real-world dataset.
- **Consensus clustering classification:** One significant contribution of this research is that it presents a new model, which uses a consensus clustering algorithm in combination with a supervised learning algorithm, to identify disease subtypes and improve predictions. The study presents a novel approach that simultaneously combines K-means clustering with DT analysis to provide a robust model that improves medical prediction accuracy by identifying sub-cohorts of patients. This approach is applied to real SSc data and evaluated. This novel nearest consensus clustering classification was created by running a K-means algorithm iteratively on different training datasets of SSc disease data. It finds consensus groups of patients based on different k-means results, and a DT algorithm is then applied to each group to classify patients' health outcomes. New patients are assigned to the group to which they are most similar and are classified accordingly.

  Details of this new algorithm model have been published as a workshop paper in Advanced Predictive Models in Healthcare and further extended and published in Journal of Healthcare Informatics Research.

- **Multilabel Classification (MLC) latent class analysis:** Another major contribution of this research is to build a novel approach combining latent class analysis with a MLC NB model to predict multiple complications by discovering hidden factors. This approach provides a robust probability model that improves medical prediction accuracy. My research presents a new algorithm, called 'Latent Class MLC Naïve Bayes', to identify subclasses and predict disease, and it was tested using SSc disease data. This algorithm first runs an iterative latent class analysis on training datasets of SSc disease data, and it finds groups of patients based on results of this analysis. Next, a multilabel NB classification is applied to each group to assign patient labels. New patients are assigned to the group to which they are most similar using a scoring formula and classified accordingly. The two new algorithms described above were tested on data from patients with SSc, and results indicate that they are more accurately in comparison to similar, alternative methods. Both models can be used to aid diagnosis and could help physicians significantly in exploring patients' characteristics, which could lead to improved PM.

Details of this algorithm have been published in a poster on International Symposium on Computer-Based Medical Systems.

- **Concept drift alleviation:** One problem ML models may face is concept drift, which occurs when a model's accuracy degrades over time. This is due to a change in the underlying relationship between the input variables and the target variables, which affects the performance of ML models. As a result, an approach that detects concept drift and updates the model accordingly is urgently needed in a healthcare context. In my study, I use the updated established drift detection method (DDM) metric to detecting drift in ML models using SSc dataset as well as a synthetic dataset of COVID-19 patients.
- **Evaluation of the proposed method's effectiveness based on both real-world clinical data and simulated data:** SSc data and simulated data, in conjunction with extensive sensitivity analyses, were used to test the new method.

1.5 Thesis organisation

The thesis aims and contributions outlined above are addressed in the next six chapters. This thesis is organised into seven chapters, each of which addresses a different aspect of the study. The contents of each chapter are described below.

- Chapter 1 provides an overview of rare diseases, ML and PM, and it identifies the problem this study aims to solve. The motivation aims and contributions of this research are also presented in this chapter.
- Chapter 2 describes previous studies on supervised and unsupervised ML methods in medicine, such as Bayesian networks, NB, DTs, artificial neural networks, ensemble learning, clustering algorithms and consensus clustering. It also describes previous studies on PM and the definition and diagnosis of rare diseases.

- Chapter 3 introduces currently available information on SSc, a rare disease, focusing particularly on symptoms and diagnosis. It details the SSc dataset used, which was provided by London Royal Hospital and includes data on 600 patients. Fundamental characteristics, implementation, and presentation of ML algorithms, such as DT, NB, K-means, and latent class analysis algorithms, along with an evaluation of these algorithms' performance, are also presented in this chapter.
- Chapter 4 proposes the first new model, which integrates a K-means algorithm with a DT algorithm in order to simultaneously deal with sampling bias, identify groups of patients who have different symptoms and outcomes and develop a transparent model for predicting those outcomes. Furthermore, it uses real clinical data and simulated data to provide a detailed definition, demonstration and evaluation of this new model and compare it to other methods. This chapter was published in the *Journal of Healthcare Informatics Research*.
- Chapter 5 proposes another new model, which integrates latent class analysis with MLC NB. This model aims to utilise latent class analysis to identify patient subgroups based on the intersection of multiple observed characteristics. Furthermore, it uses both real clinical data and simulated data to offer a detailed definition, demonstration and evaluation of this new model as compared to other methods. This chapter was published for *the Computer-Based Medical Systems (CBMS) conference.*
- Chapter 6 provides detailed information regarding the problem of concept drift and focuses on the implementation of the proposed DDM metric for detecting drift in ML models using both an SSc dataset and a synthetic dataset of COVID-19 patients.
- Chapter 7 conclude the study and summarises all key achievements. This chapter presents the results of this work and provides recommendations for future research.

1.6 Summary

This chapter provided an overview of the current research, including its aims and motivation, and it described the need for this study. It also presented the contributions of this research and a thesis road map. The following chapter describes previous studies on supervised and unsupervised ML in medicine as well as on rare diseases and PM. Moreover, it addresses previous studies on concept drift.

Chapter 2 Literature Review

2.1 Introduction

As discussed in the previous chapter, ML can be a powerful tool for diagnosing diseases, predicting patient health outcomes and improving PM. This chapter summarises the literature on ML and its current applications in the medical field. The first half of this chapter explains and examines ML techniques used in medicine, including supervised and unsupervised learning methods, and it offers a critical assessment of their advantages and disadvantages. Additionally, it presents resampling techniques for generating random samples from a dataset. The second half of this chapter defines uncommon and rare diseases and analyses the difficulties associated with detecting and treating them. Furthermore, it presents the term ‘personalised medicine’ (PM), which is used to treat patients more effectively. This chapter also provides background information on concept drift, which affects ML models over time.

2.2 Machine learning in medicine

Generally speaking, the ML field is based on a computer’s ability to learn from a given dataset and perform complex tasks. The aim of ML is to design and develop algorithms using computational techniques based on previous data in order to solve real-world problems. Utilizing vast amounts of raw data, these algorithms provide meaningful knowledge, which can be used to complete tasks efficiently and reach meaningful decisions. ML methods in medicine attempt to limit human decision-making and do more than physicians would be able to do alone (Deo, 2015). Therefore, ML methods can be used to assist with disease diagnosis, health outcome predictions and decision-making, thereby leading to improved healthcare systems. For instance, skin cancer may be predicted and categorised using ML algorithms, and the application of ML has aided in the prediction of disease development from pre-diabetes to type 2 diabetes (Sidey-Gibbons, 2019).

Vayena et al. (2018) argued that incorporating ML into the medical sciences field has become increasingly essential due to the existence of vast medical datasets that physicians are unable to utilise effectively (Vayena et al., 2018). Additionally, physicians tend to assume that a patient’s problem is associated with their field, even when it is not. For example, Saarela et al. (2019) stated that, if a patient with pain in their arm is brought to a cardiologist for treatment, the cardiologist is likely to presume that the patient has a coronary issue. However, the same patient may be diagnosed with a cervical disc disorder if they are referred to a neurosurgeon. The actual disease may be detected eventually, but it may take a long time to arrive at the final, correct diagnosis. Thus, an efficient ML system could assist physicians in accurate decision-making (Saarela et al., 2019). ML in medicine plays a significant role in improving medical performance and making healthcare systems more efficient; in the future, for instance, patients could use ML medical systems for self-management via their smart devices (Vayena et al., 2018).

There two main types of ML methods: supervised and unsupervised learning. These are described in the following sections. Supervised learning methods learn from labelled data to predict outcomes based on training examples, whereas unsupervised learning methods learn from unlabelled data to analyse and cluster patients into groups (Deo, 2015).

2.2.1 Supervised Methods

There are three types of ML algorithms: supervised learning, unsupervised learning, and reinforcement learning (see Figure 2.1).

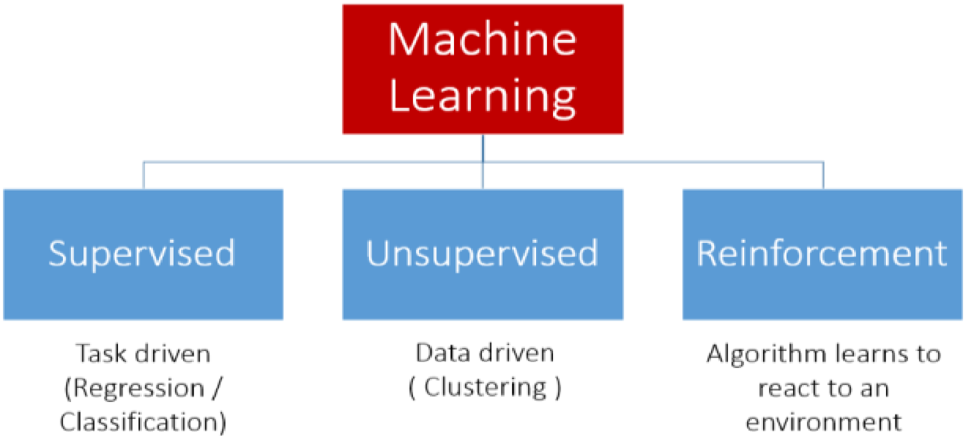

Machine Learning
Supervised
Unsupervised
Reinforcement
Task driven (Regression / Classification)
Data driven (Clustering)
Algorithm learns to react to an environment

Figure 2.1: Types of Machine Learning (Praveena & Jaiganesh, 2017).

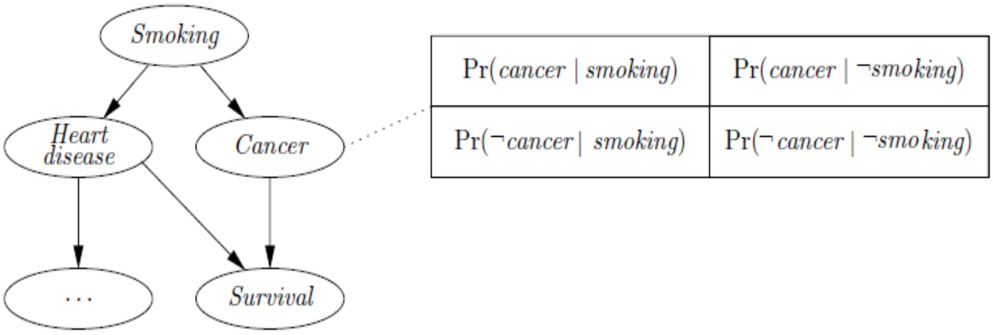
Supervised learning algorithms are trained from datasets in which each training record has an input value and a predetermined output value. A supervised learning algorithm learns to develop a model that identifies links between the input and output values for each training example, thereby enabling it to predict the output value corresponding to any new input value (Praveena & Jaiganesh, 2017).

Classification and regression algorithms are common supervised learning models. This study focuses on classification approaches because classification models have been proven to be beneficial when applied to biological problems. Classification algorithms such as NB, DT, and random forests, for example, have been successfully employed as models to predict plant virus encoded RNA silencing. Classification algorithms have also been used as significant ML tools for the diagnosis, analysis, and clinical management of eye disorders (Jagga & Gupta, 2014). Although supervised learning models can be used to address various medical problems, the exact choice of algorithm depends entirely on the nature of the problem to be solved. In general, neural networks are effective for assessing continuous variables, while DT and NB algorithms are more effective for analysing discrete variables. The latter two are also transparent models and have been used effectively in the healthcare domain (Kotsiantis et al., 2007). For medical diagnosis, a variety of supervised learning approaches are available, and new strategies are constantly being developed in order to overcome the limitations of traditional models. These innovative strategies are usually intended to find a more accurate solution to a problem and to aid professionals in their diagnoses. The main supervised learning methods are discussed in the following section.

2.2.1.1 Bayesian Networks (BNs)

Lucas (2001) noted that the 1990s saw an upsurge in researchers attempting to develop medical applications using ML methods. Bayesian Networks (BNs) were at the forefront of research in this area. The BN methodology was proposed to help develop a realistic model of a medical disorder. BN models operate despite uncertainties involved in the medical science field and, particularly, the diagnosis and treatment selection processes. BNs are structured by medical specialists, who compute probabilistic statements for the variables involved to determine the relationships among them.

When using BNs, specialists determine the probable relationships among medical factors used to diagnose a disease, and symptoms, signs, tests, and scans are used to confirm a diagnosis (Gadewadikar et al., 2010). The general structure of a BN is displayed in Figure 2.2, using the health risks of smoking as an example.



$\Pr(cancer \mid smoking)$	$\Pr(cancer \mid \neg smoking)$
$\Pr(\neg cancer \mid smoking)$	$\Pr(\neg cancer \mid \neg smoking)$

Figure 2.2: An Example of a Bayesian Network (Lucas et al., 2004).

Decision-making, casual knowledge, and existing expert knowledge are necessary to build BNs. BNs are developed using readily available data, which implies that human experts' knowledge and opinions are not needed. However, to ensure effectiveness and accuracy, data collection must be conducted carefully. All values and variables present in the data should match those in the modelled network. BNs can be developed easily by hand and then utilised extensively in biomedicine and the healthcare domain. A BN is created based on the relevant information from physicians and patient data and is represented by a graph consisting of variables and their conditional interdependencies is created. Each node on the graph denotes medical goals, evaluations, and clinical symptoms, and the structure of the graph assists in measuring their relationship. Therefore, developing a BN from data requires network structures along with learning parameters, both of which demonstrate conditional probability (Lucas et al., 2004).

One advantage of a BN structure is its flexibility and the fact that it can be easily understood by healthcare professionals (Lucas, 2001). BNs can be used to effectively predict breast cancer because they identify the relationship among diagnoses, test results, and imaging studies. Furthermore, BNs have been used to create a user-friendly webpage that predicts the initial diagnosis of Alzheimer's disease, which lent support to medical decision-makers. In conclusion, BNs, which are probabilistic graphical models, are rich frameworks for use in medical diagnosis (Alexiou et al., 2017).

2.2.1.2 Naïve Bayes (NB)

Like BNs, the Naïve Bayes (NB) supervised learning method can assist medical professionals with treatment and diagnosis. Several supervised learning systems that use an NB algorithm to forecast disease and improve physician decision-making capacities have been used successfully in medicine. Moreover, NB algorithms have reportedly performed better than other classification methods in medical settings (Bohra et al., 2017). Wei et al. (2011) asserted that NB algorithms show a heightened level of accuracy when the variables are highly related or independent. Moreover, researchers have also found that the combination of NB and unsupervised learning in medicine leads to increased accuracy (Wei et al., 2011). Kharya et al. (2014) provided a graphical user interface for entering patients' screening records and detecting their probability of having breast cancer using NB classifiers. Results showed that NB classifiers improved accuracy, required little processing effort and operated quickly (Kharya et al., 2014). Moreover, when Chaurasia et al. (2018) designed three models for predicting breast cancer, their results showed that, of the algorithms, the NB algorithm performed the best, with a classification accuracy of 97.36% (Chaurasia et al., 2018). NB algorithms have also been applied to create a model that can accurately predict the likelihood that a patient has diabetes. Results showed that NBs perform with an accuracy of 76.30% (Sisodia & Sisodia, 2018). Gupta et al. (2020) developed six classification models to predict coronary artery disease; the NB model performed with an accuracy of 88.16% on the test set (Gupta et al., 2020). Moreover, an NB algorithm was used to effectively predict the spread of swine flu, one of the most highly infectious diseases. NB models are transparent and assist physicians in making diagnoses (Srinivas et al., 2020). Thus, it may be concluded that NB models improve and enhance the decision-making process and can be used to effectively address serious questions regarding diseases, such as their diagnosis and treatment.

2.2.1.3 Decision Trees (DTs)

Decision trees (DTs) are supervised ML methods that have been used extensively in medical decision-making due to their effectiveness and reliability. Decision-making by humans alone has become increasingly complex due to the vast amounts of data that must be analysed. Therefore, the need for a good decision support system has arisen (Podgorelec et al., 2002). A DT has been used successfully to diagnose renal calculi.

In general, this disease, wherein the kidneys fail to dispose of some of the body's waste, is most likely to affect people in their 30s. The DT shows that the amount of erythrocyte in the urine is the key indicator of renal calculi. If this amount is greater than 10 RBC/µl, this means that the risk of renal calculi exists. If this amount falls below 4, which is rare, then the risk is extremely low. However, if it is between 4 and 9, which is normal, urine colour must be examined. If it is yellow, the risk is low (Topaloglu & Malkoç, 2016). A DT algorithm has also been used to predict coronary heart disease. The performance of this algorithm was good, with a success rate of 69.51% (Kim et al., 2015). Moreover, a DT algorithm has also been applied to predict axillary lymph node mitosis in primary breast cancer. This model displayed a high level of accuracy when clinical variables were used, implying that the DT model might be able to assist oncologists before the start of treatment (Takada et al., 2012). Madadipouya (2015) asserted that DTs are beneficial in the medical field as a result of the advantages they possess. Because they offer the clarity doctors require, they are considered one of the most effective models for use in medicine. In other words, medical decisions must be made effectively and reliably, and a simple decision-making model such as a DT is useful and provides a high level of accuracy (Madadipouya, 2015). Podgorelec et al. (2002) also noted that DT methods are reliable, effective decision-making methods that physicians can easily understand; furthermore, despite their simplicity, they provide high classification accuracy within the medical field. Many medical studies have applied DTs as supervised learning methods to classify and diagnose diseases using real medical datasets. These studies have shown that DT models can play a significant role in facilitating the accurate diagnosis of diseases (Podgorelec et al., 2002).

2.2.1.4 Artificial Neural Networks (ANNs)

An artificial neural network (ANN) is a classification method that mimics how the human brain analyses and processes information. It is applied to problems that would be difficult for humans to solve.

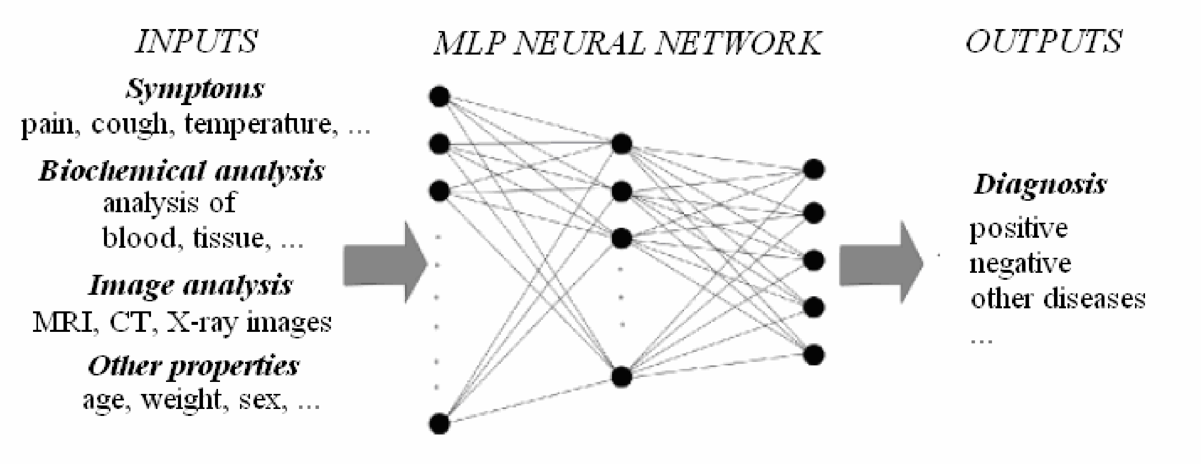


Figure 2.3: Medical Diagnostic Example Using a Neural Network (Kajan et al., 2014).

The neural network method is a non-linear mathematical model that comprises three distinct layers: the input layer, the output layer, and the hidden layer (Kajan et al., 2014). The major benefits of using this method are its high accuracy, easy maintenance, and high noise tolerance. Its application to heart failure prediction showed that it performs significantly better than traditional statistical methods (Akhil Jabbar et al., 2012). It has been reported that this type of network provides considerable relief to doctors who work under extreme pressure in emergency departments (Falavigna et al., 2019). Although neural networks are able to outperform nearly all other classification methods, they are considered ‘black box’ methods; that is, it is very difficult to understand how results are actually obtained. By contrast, DT algorithms are interpretable models used widely in medical domains.

2.2.1.5 Ensemble Method

In addition to the above methods, the ensemble method is the combination of multiple classifiers in order to obtain better predictive performance. The combination of various sets of ML methods helps to improve results and solve a given problem (Chakraborty, 2017). The following diagram (Figure 2.4) shows that the ensemble method, as a combination of learning algorithms, could provide a better solution to medical models than standard methods. Briefly, instead of using one single learning algorithm to solve a problem, the ensemble method takes multiple learning algorithms into account. Thus, the primary aim of this method is to convert weak learning algorithms into strong ones (Lo et al., 2008).

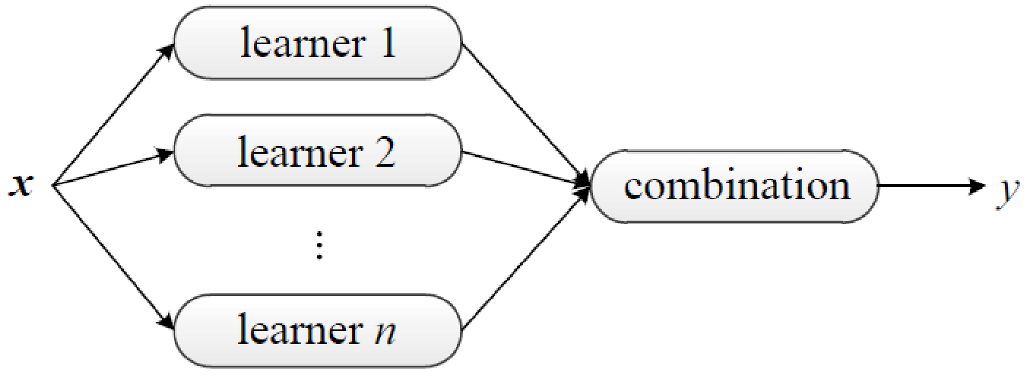


Figure 2.4: Common Ensemble Architecture

Numerous ensemble learning algorithms have yielded comparatively high levels of accuracy in medical settings. Jain et al. (2000) asserted that the main benefits of combining multiple classifiers are decreased variance, decreased bias, and improved predictions. Hence, any ensemble method that creates various training sets and includes different classification methods could help to improve classification accuracy and produce better results (Jain et al., 2000).

The most common ensemble algorithms are bagging, AdaBoost, and mixtures of experts. These are described below.

- Bagging**:** This algorithm generates different training datasets, called bootstraps, by using a sampling technique. Predictions are generated either through uniform averaging or through voting over class labels (Brown, 2010).
- AdaBoost**:** In this algorithm, a variation of bagging, each individual model is analysed sequentially depending on the previous model, and parameters are adjusted during each iteration to correct errors until the final model is created (Brown, 2010).
- Mixtures of experts**:** Mixtures of experts is a popular and intriguing ensemble strategy that has much potential for improving ML performance. Based on the divide-and-conquer approach, it involves dividing the problem space into subtasks and using expert models to complete each (Masoudnia & Ebrahimpour, 2014).

2.2.2 Unsupervised Methods

In addition to supervised learning, unsupervised learning is the other primary type of ML used to analyse and cluster unlabelled data. Unsupervised learning methods are proving to be promising in medicine; they have been used to identify different phenotypes of certain diseases and patterns within medical datasets (Guan et al., 2016). Lopez et al. (2018) presented an unsupervised clustering algorithm to identify groups of patients based on their genomic makeup, and they discovered that this approach aided in the advancement of PM by identifying the most characters for each patient group (Lopez et al., 2018).

2.2.2.1 Clustering Algorithms

One primary aim of unsupervised learning is to discover hidden patterns in unlabelled data in order to identify similarities among patients. Therefore, cluster analysis is one of the main methods used in unsupervised learning. In medicine, cluster analysis is a set of methods used to divide patients into various groups. Each patient in a group shares similar characteristics with the others in the same group. K-means clustering, hierarchical clustering, and DBSCAN clustering are crucial clustering techniques (Lütz, 2019).

The K-means algorithm is simple and straightforward. It divides a dataset into *K* different groups. First, *K* initial centroids are selected at random, and each point is assigned to its closest centroid (Pandey et al., 2013). Collection of points assigned to a specific centroid are grouped as a crystal. An advantage of this method is its simplicity, and a disadvantage is the fact that this approach requires the number of clusters to be known in advance (Nithya et al., 2014).

There are two types of hierarchical clustering: agglomerative algorithms (bottom-up) and divisive algorithms (top-down). The former approach works by merging similar clusters, while the latter divides data into subclusters. The procedures associated with these two approaches are diametrically opposed. Although hierarchical clustering has the advantage of flexibility, it also has a disadvantage in that its termination criteria are vague (Nithya et al., 2014).

DBSCAN is a density-based clustering algorithm. It combines data points located within the same area. Hence, all the points in a dense region are clustered together (Ratnawati et al., 2018).

Because clustering algorithms are unsupervised learning methods, it is unclear which method is best suited for a particular task; thus, determining the quality of clustering algorithm results is an important task. In general, the metrics for evaluating clustering algorithms (cluster validation) are either internal validation methods, which measure the quality of a clustering algorithm without external information, or external validation methods, which measure the quality of a clustering algorithm based on its results, an external source and relative metrics (Tan et al., 2006).

Using clustering evaluation metrics, Kalyani (2012) discovered that the use of clustering algorithms, especially the K-means method, is effective in the medical sciences field. The K-means method has also been used to effectively predict heart disease, which has provided significant assistance to healthcare professionals (Kalyani, 2012). Moreover, when the K-means method was compared to a DBSCAN clustering algorithm using a healthcare dataset, the former performed better in terms of both accuracy and execution time (Ogbuabor & Ugwoke, 2018). Ren and Wang (2018) conducted a comparative analysis of three clustering algorithms (K-means, DBSCAN, and hierarchical clustering) using medical datasets. Results showed that the K-means method performed better than both DBSCAN and hierarchical clustering. Furthermore, Lütz (2019) performed various experiments on an online breast cancer dataset using Weka software; results demonstrated that the K-means method can be used to effectively predict breast cancer (Lütz, 2019). Rajalakshmi et al. (2015) utilised the K-means algorithm to effectively predict chronic diseases such as heart disease, liver disease and cancer (Rajalakshmi et al., 2015).

Not only has the K-means algorithm’s significance for medicine been demonstrated, but also researchers have suggested that integration of the K-means method with another method could improve the quality of the model. Ratnawati et al. (2018) proposed a new approach combining K-means clustering with NB classification to predict the early stages of cancer and proved that this integration produces effective results. According to their results, the K-means method can be used to predict breast cancer with 91% accuracy, and the integration approach provides 95% accuracy

(Ratnawati et al., 2018). Additionally, a consensus clustering model that uses multiple clustering algorithms may be more accurate than other clustering algorithms in medicine.

2.2.2.2 Consensus Clustering

Consensus clustering, or ensemble clustering, integrates various clustering methods in order to produce a robust model. These are utilised as the input source for a given dataset in order to achieve consensus-based clustering. This model outperforms all individual clustering algorithms, demonstrating that the combination of results is more consistent than clustering algorithm results. The main advantages of this approach are its robustness, consistency and novelty (Goder & Filkov, 2008). Figure 2.5 depicts the main structure of this model.

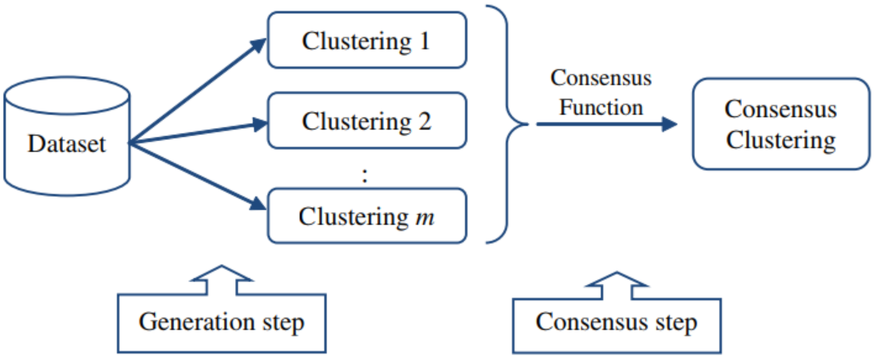


Figure 2.5: The General Process of a Cluster Ensemble (Vega-Pons & Ruiz-Shulcloper, 2011).

As shown in the above figure, a consensus clustering algorithm is constructed in two steps:

1. Generation: a number of clustering methods is combined.
2. Consensus: the function method is selected.

Tucker et al. (2016) utilised the consensus clustering method to identify the subclasses of a certain disease. They found that the successful determination of disease subclasses assists in the diagnosis and outcome prediction of that disease (Tucker et al., 2016). Liu et al. (2017) utilised the consensus clustering model within a medical department and, using 110 synthetic datasets, proposed a unique algorithm called entropy-based consensus clustering for patient stratification. The performance of this algorithm was unsurprising; it showed better results than individual clustering algorithms alone (Liu et al.,2017). Lourenço et al. (2014) tested the consensus clustering algorithm to analyse electrocardiography (ECG) in patients. Nowadays, ECG assists greatly in disease diagnosis (Lourenço et al., 2014).

Consensus clustering is one of the most important tools in medicine, especially for the discovery of subclasses within medical data. The current study was conducted using a consensus clustering algorithm, as described in Chapter 4. The performance of all preceding methods may be enhanced when resampling approaches are applied.

2.3 Resampling

Resampling is a statistical strategy that depends on empirical analysis based on actual data, rather than on asymptotic and parametric theory. The aim of resampling is to arrive at an inferential decision. ML models might use resampling methods to improve model performance (Beasley & Rodgers, 2009). Dodangeh et al. (2020) combined ML models with resampling methods when conducting flood susceptibility prediction. He found that resampling algorithms like bootstrapping and subsampling increased the models' performance (Dodangeh et al., 2020). Many medical outcome variables, such as survival status and the presence of disease indicators, are dichotomised. The binary values of an outcome-dichotomised variable are called 'classes'. A class imbalanced problem occurs when a medical dataset with binary outcomes consists largely of one class. In this case, ML models prioritise correctly categorising the large class while misclassifying the small class. Resampling techniques such as over-sampling, under-sampling, bootstrapping, and cross validation can be used to solve this issue (Lee, 2014). Shi et al. (2022) proposed a resampling method to improve the prognostic model of kidney disease. He aimed to offer a resampling strategy to address the predictive model's unbalanced data structure problem and enhance its predictive performance (Shi et al., 2022). Hence, all preceding methods include resampling techniques to address the difficulties associated with rare disease diagnosis, as described in the next section.

2.4 Rare diseases

Rare diseases refer to frequent, persistent, and continuous deadly medical condition that affect few people compared with other conditions. They impact about 6% of the world's population. Accessing suitable treatment choices is difficult for many patients with rare diseases. These rare diseases are treated with orphan drugs. Varying terminology and conflicting definitions of rare diseases are regarded as significant barriers to treatment accessibility. "Rare diseases" does not have a universal definition. An extensive variety of international definitions for rare diseases have been proposed and implemented. This is due to the fact that diverse groups of individuals have differing ideas and worries on the same subject. Decision-makers, patient groups, regulatory agencies, industry, reimbursement bodies, payers, policymakers, and scientific organisations are examples of stakeholders. Payers are primarily concerned with the costs and advantages connected with rare diseases, whilst patient advocacy organisations prioritise treatment accessibility. Furthermore, policymakers view rare diseases as improving the effectiveness of the health system and healthcare delivery. Furthermore, the criteria employed in defining rare diseases differ between organisations and countries. Some definitions employ qualitative norms with emotional linkages, including disease severity or alternative treatment availability. According to the UK's Rare Disease Framework (2021), rare diseases are those that impact fewer than 1 in 2,000 people. The EU defines "rare diseases" life-threatening or chronically debilitating disorders with a low prevalence (less than 5 per 10000) that require coordinated measures to prevent significant morbidity (Abozaid et al., 2022).

Between 5000 and 8000 different rare diseases have been described. Also, many rare diseases are routinely published in the medical literature. Therefore, it is essential that different stakeholder groups understand the terminology related to definition of rare diseases. The lack of a universal definition of rare disease led to the increased colloquial application of terms such as neglected and ultra-orphan in a manner that may not conform to their formal descriptions. Richter et al. (2015) provided an overview of the terminology utilized in the definitions of rare diseases and the health technologies associated with them through Web search from 32 relevant national and international organization. According to their investigation, the term "rare diseases" has the most frequent usage, with 112 definitions out of 296 distinct definitions. In addition, 172 of 296 definitions provide a prevalence threshold. Also, the research indicated that "rare disease" is the preferred term because it was used in more than six times as many definitions as "orphan disease". The fact that the phrase "orphan drug" appeared in numerous definitions shows that it is most frequently used to describe medical advancements made to treat rare disorders. They concluded that there is a universal preference for using the phrases "rare disease" and "orphan drug" when defining a rare disease and the accompanying technology. Although the majority of definitions include a prevalence threshold, there are few criteria relating to illness severity and the absence of available treatments (Richter et al.,2015).

Rare diseases affect a small percentage of the population. In general, such diseases affect fewer than one to five of every 10,000 people. Currently, the majority of rare disease patients do not receive effective medication. Additionally, such patients experience support issues because it may take a long time to receive a diagnosis. Worldwide, 350 to 400 million individuals are affected by uncommon diseases, 80% of which are genetic. Lack of diagnosis, absence of medical skills, absence of accessible treatment and limited data availability are the primary difficulties associated with the study of uncommon diseases (Peberdy, 2017). Any delay in the diagnostic procedure could affect patient survival. Moreover, gathering information on rare diseases is difficult due to a lack of resources and the low number of patients affected (Rodwell & Aymé, 2015). Additional obstacles when attempting to diagnose rare diseases include the fact that some patients may have mild symptoms and never visit their general practitioners, and some may die before receiving a diagnosis (Black et al., 2015). Rare Disease UK stated that most rare disease patients receive little information from physicians before and after diagnosis and only receive such information when they connect with other patients who have become experts on their disease. Some patients who receive a diagnosis after being seen by many specialists may actually receive an incorrect diagnosis, and those who have not yet received a diagnosis may find it difficult to exercise, which may result in depression. In addition, half of uncommon disease patients are youngsters, meaning that their families also suffer (www.raredisease.org.uk).

Figure 2.6: Fewer than One in Ten Patients with Rare Diseases Receives Disease-Specific Treatment (Gammie et al., 2015).

Svenstrup et al. (2015) asserted that the nature of rare diseases and the continuous delay in their diagnosis makes treatment challenging. They also discovered that most physicians preferred to use a medical decision-support system when diagnosing rare diseases because of the expert knowledge required. Further, the researchers proved that the use of these decision-support systems to identify rare diseases reduced human error and the chances of providing an incorrect diagnosis (Svenstrup et al., 2015). MacLeod et al. (2016) utilised an ML algorithm to identify rare diseases based on a behavioural dataset using the functional gradient boosting approach. They found that rare disease patients face unique challenges compared to patients with chronic diseases. For instance, rare disease patients join health support groups, search for information, and watch videos to attempt to understand their disease, while patients with chronic diseases generally do not (MacLeod et al., 2016).

ML methods could improve the diagnosis and the treatment of rare diseases. Indeed, advanced ML methods have become a crucial part of research as they help to reduce diagnostic error and provide consultants with decision support (Soni et al., 2018). ML is transforming medicine and healthcare, and it has the potential to improve the detection and treatment of rare diseases (Schaefer et al., 2020). Chernbumroong et al. (2020) applied an unsupervised ML method to predict disease manifestations and outcomes in lymphangioleiomyomatosis, a rare multisystem disease. This method revealed clinically significant clusters linked to complications and outcomes, and its use improved decision making and patient prognosis (Chernbumroong et al., 2020). This thesis focuses on rare disease analysis using advanced ML methods in order to improve the accuracy of health outcome predictions and personalised treatment.

2.5 Personalised medicine (PM)

The promise of using genomes to create treatment that is more precisely tailored to the unique biology of individuals and the diseases they suffer from has generated optimism, ambitious goals, and substantial investment. This has been motivated by the concept. Since the beginning of clinical practise, physicians and other medical professionals have wished for the capacity to personalise care and treatment to the individual needs of each patient. It is no secret that personalised medicine has been on the rise in recent years, bringing with it both money and a healthy dose of hope and scepticism from the general public. Despite the fact that the word “personalised medicine” (PM) has been interpreted in a variety of ways, several efforts have been made to clarify its actual meaning. Schleiden et al. (2016) attempted to clarify the concept through a systematic review accurately. The attempt seeks to address the fundamental need for guidance through the argument. Schleiden and colleagues provide the definition based on solid evidence and claim the vital practical implications. To allay unfounded public concerns and expectations, and to prevent interested parties from placing their own interests ahead of those of the patients they serve, it is crucial to persuade legislators to consider the patients’ best interests when formulating regulatory strategies and making “policy decisions”; to allay unfounded public concerns; and to allay unfounded public expectations (Grandis & Halgunset, 2016).

The term “individualised medicine,” sometimes known as “personalised medicine” (PM), has gained popularity in both academic and popular discussions of health care. The problem is that PM is not well-defined and can be understood in different ways. This theoretical ambiguity complicates the public discussion of PM’s potential, risks, and limitations. Schleiden et al. (2013) provided a systematic literature review to determine how PM is utilised in current scientific practices. Using PubMed, they searched for “individualised medicine” and “personalised medicine” as key words. They located 2457 works with “PM” in the title or abstract. It has been found that the growth rate of literature on PM was 49% annually on average. Moreover, the statistics demonstrate that there is no consensus regarding the definition of PM. It’s worth noting that the word “PM” seems to be used in a variety of contexts within the healthcare system, including direct patient care, scientific investigation, and the approval of new medications. The objective of personalised medicine is to determine the appropriate treatment for each patient in order to optimise treatment benefit and minimise unwanted effects. Schleiden et al. (2013) concluded precisely based on their findings that PM attempts to enhance the timing and stratification of healthcare by applying biological data and biomarkers in accordance with genetics, metabolomics, proteomics, and molecular pathways (Schleiden et al., 2013).

Former US president Barack Obama launched the Precision Medicine Initiative in 2015. Since then, precision medicine, also called PM, has been in the national spotlight in the US. This medical model attempts to focus on each patient individually instead of on the population more generally, and treatment is based entirely on personal patient information (Nimmesgern et al., 2017). The PM model, which is an essential subject in the medical field, considers all patient details such as lifestyle, clinical factors, and genetic factors to make an informed decision on patient health (König et al., 2017). Johnson et al. (2020) stated that precision medicine is growing at a comparable rate and is likely best defined as a healthcare movement. It enables healthcare practitioners to uncover and provide information that either supports or adjusts the trajectory of a medical decision based on individuals' unique characteristics. Thus, clinicians provide individualised care to each patient, which was not previously possible (Johnson et al., 2020). Researchers have developed ways to evaluate, integrate, and interpret the large amounts of data generated by high-throughput, data-intensive biomedical research assays and technology. Though numerous statistical approaches have been developed to accommodate vast quantities of data using AI techniques, there is a need for ML models to adjust, or 'personalise', medications based on the complex and frequently unique characteristics of certain individuals (Schork, 2019).

The use of artificial intelligence (AI) to analyse genomic data and then use that information to tailor therapy to each individual patient is a particularly intriguing use of precision medicine. Therefore, the field of personalised medicine is one that is constantly evolving as medical professionals gain knowledge on the best ways to use diagnostic tests to identify which treatments will work best for specific patients and how to use medical interventions to change biological systems that have an impact on health (Pelter & Druz, 2022). “Artificial intelligence informed care decisions are sort of personalisation that is garnering attention and investment”. However, this form of personalisation is not traditionally considered to be a component of personalised medicine. For machine learning, a broader range of health and care data is required, likely including the entire “electronic health record (EHR) and more patient-generated monitoring and lifestyle data. Access to precise health data is the next obstacle for scaling personalised treatment. To guide the development of new targeted medications, to identify unmet needs, and to assess the value of novel therapies", it is necessary to collect new information from vast data populations (Kalra, 2019).

Precision medicine, in which a patient receives medical care and therapy based on their particular disease profile, is one of the most promising application areas for ML. Precision oncology, in which the objective is to prescribe cancer treatments based on the genetic characteristics of a tumour, is a classic example of the problems with and prospects for ML in precision medicine. ML promises a future of rigorous, outcomes-based medicine, with detection, diagnostic, and treatment procedures that are constantly tailored to individual and contextual variances (Goecks et al., 2020). Briefly, ML can play an important role in PM by improving disease diagnosis and patient treatment. For instance, clustering methods, in which the most similar patients are clustered within one group, helps to personalise medicine. Hence, the stronger the clustering method, the better the PM model.

2.6 Concept drift

As mentioned in the preceding chapter, concept drift is a crucial problem in ML because it results in a significant performance decrease in the model over time. Concept drift occurs when the statistical features of data vary over time, thereby reducing the accuracy and efficacy of trained models. Therefore, it is critical to understand existing DDMs in order to identify the hazards associated with them and suggest robust solutions to these problems (Hashmani et al., 2020). In healthcare, clinical profiles are dynamic; the underlying data distributions that characterise patients can vary over time (data drift), as can the relationship between input features and clinical outcomes (concept drift). Thus, current algorithms must be monitored regularly to verify their safety. The advent of the COVID-19 pandemic, which has caused a significant, rapid, and continuing shift in conditions across industries from financial services to healthcare, is a prime example of data drift. In April 2020, the United Kingdom's National Health Service (NHS) recorded a 57% decrease in emergency department (ED) attendance, corresponding to 120,000 fewer ED attendances in April 2019. During the first wave of the pandemic, (March–May 2020), the exponential growth in COVID-19-related attendances necessitated drastic adjustments to operational procedures (Duckworth et al., 2021).

Müller & Salathé investigated the impact of ML concept drift by focusing on Twitter attitudes about vaccines, a topic of critical importance during the COVID-19 pandemic. They found that, due to concept drift, models trained on pre-pandemic data would mostly have failed to identify the decline of vaccine sentiment during COVID-19. They suggested that social media analysis systems must continuously address concept drift in order to prevent a potential decrease in model performance (Müller & Salathé, 2020). In the healthcare industry, concept drift of ML models can result in poor decision-making; thus, there is a need for concept drift approaches that maintain accurate ML performance. Concept drift and methods used to address it are explored in detail in Chapter 6. In addition, this study presents an established DDM implemented on SSc and synthetic COVID-19 datasets.

2.7 Summary

This chapter reviewed existing studies related to ML in medicine. It described in detail the concept of ML and its various types – supervised (e.g. DT and NB) and unsupervised (e.g. K-means and consensus clustering) learning – in medicine. Moreover, this chapter presented the concept of rare diseases and described the use of ML methods to analyse and diagnose them as well as to provide PM. Additionally, this chapter provided a brief explanation of concept drift and its effect on ML models.

This chapter revealed that classification methods can be used to diagnose a disease and that consensus clustering methods may yield better, more accurate results than such methods used individually. Based on the points above, this research focuses on the combination of classification methods via clustering methods to accurately diagnose rare diseases. This study examines the consensus clustering method, which divides patients into robust subgroups and uses classification methods to predict patient disease within each group. Additionally, this study utilises an established DDM algorithm implemented on SSc and a synthetic dataset of COVID-19 patients to maintain ML model performance.

The following chapter explores SSc, a rare disease. The implementations of standard classification and clustering methods are also presented in the next chapter.

Chapter 3 Preliminaries

3.1 Introduction

SSc is a rare and potentially fatal illness that affects the skin and other organs of the body, such as the blood vessels, muscles, heart, lungs, and kidneys. It remains difficult for physicians to diagnosis due to the paucity of information and the limited number of patients. As a result, advanced ML algorithms are being considered to support the decision-making of physicians. This chapter provides an overview of SSc, exploring causes, types, clinical features, symptoms, diagnosis, and prognosis. In addition, a dataset of SSc patients obtained from the Royal London Hospital is discussed. Furthermore, this chapter illustrates the fundamentals of ML methods – such as DTs, RFs, NBs, K-means, and LCA – which may help to diagnosis SSc. A DDM method that monitors the model over time is also described. Finally, this chapter explains how ML models can be evaluated.

3.2 Systemic Sclerosis

3.2.1 Definition

Scleroderma diseases can be divided into two main types: localized scleroderma and SSc. Localized scleroderma affects mainly the skin without visceral organ involvement while SSc can affect the entire body. SSc is a rare, multisystem autoimmune illness characterised by skin and internal organ fibrosis and vasculopathy. It is a rheumatological disease with a high death rate and significant consequences. The major complication of this disease is renal crisis, which may culminate in death from malignant hypertension and renal failure. Mortality may also occur due to the complications of lung disease, pulmonary fibrosis, and pulmonary hypertension. Survival rates in the United Kingdom have improved, in part due to the availability of specialized centres. SSc pathogenesis involves small vessel vasculopathy, the production of autoantibodies, and fibroblast dysfunction, with skin thickening as the most common symptom. SSc is more prevalent in women, who are four times more likely to develop SSc disease (particularly in the third and fourth decades of life), although mortality is greater in men (Bosoni et al., 2016). The United Kingdom and Japan report that the prevalence of this disease is around 35 cases per 1 million adults. In the United States, it has an annual incidence of about 20 cases per 1 million. Age, gender, and ethnicity are the principal factors that contribute to disease susceptibility. SSc patients require comprehensive diagnosis and follow-up, especially as treatment must be tailored to organ presentations (Becker et al., 2019). Hence, better understanding and management of SSc have resulted in better disease care, including better classification and more systematic assessment and follow-up (Denton & Khanna, 2017). Bonomi et al. (2022) mentioned that machine learning (ML) has been applied to the classification of SSc patients in order to identify those at high risk of developing major complications. Also, ML may be useful for early detection of organ involvement (Bonomi et al,2022).

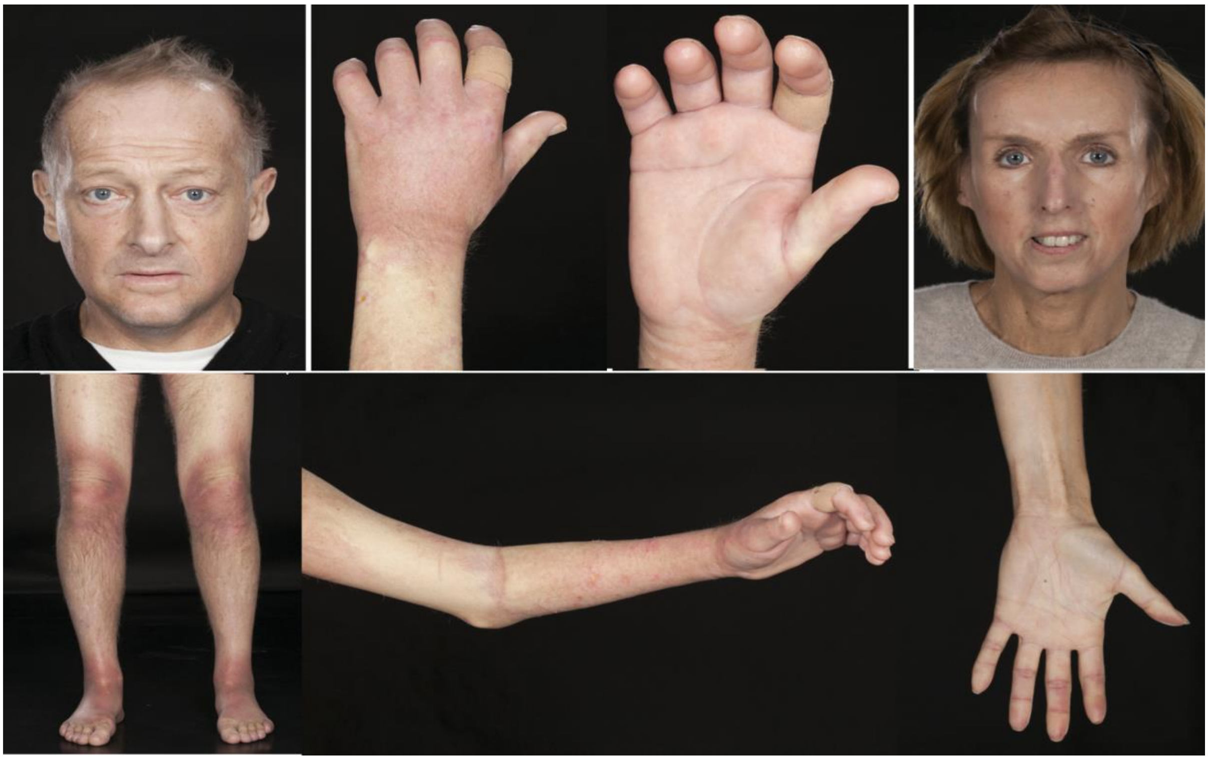


Figure 3.1: Diffuse Cutaneous Systemic Sclerosis (Denton & Khanna, 2017).

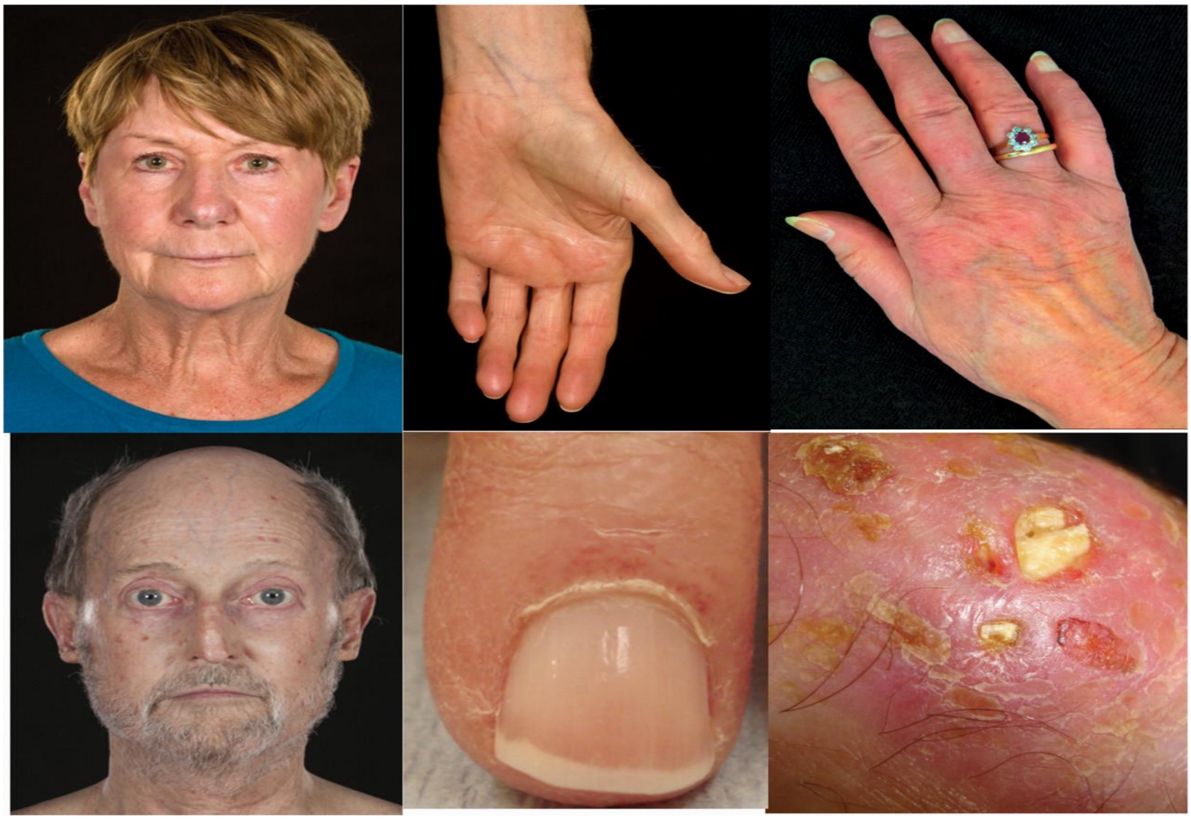


Figure 3.2: Limited Cutaneous Systemic Sclerosis is Associated with Mild Skin Involvement Distal to the Elbows and Knees, With or Without Face and Neck Involvement, and Sparing of the Chest and Abdomen(Denton & Khanna, 2017).

3.2.2 Subsets and Clinical Features

SSc can be categorised as two subgroups: limited cutaneous SSc (lcSSc; the most common type), which affects the skin distal to the elbows and knees, and diffuse cutaneous SSC (dcSSc), which affects the extremities proximally and distally. However, both SSc subgroups share common features, such as Raynaud's phenomenon, heartburn, skin sores, abdominal grumblings, and other organ involvements (Allanore & Distler, 2015). The presenting symptoms may differ, and the subgroups have different prognoses.

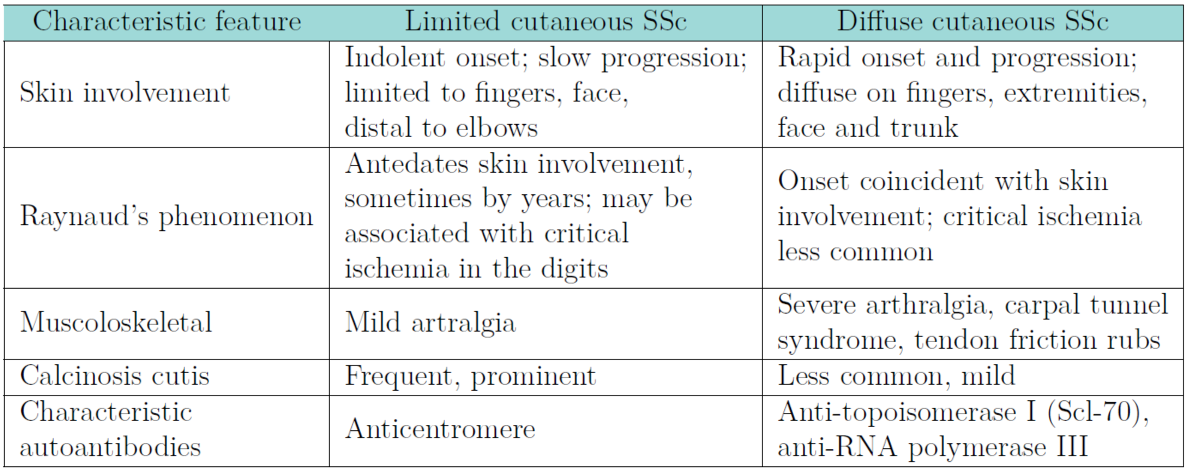
The clinical features of SSc derive from the combination of fibrosis and vascular abnormality, thus SSc is not primarily an inflammatory disease. Raynaud's phenomenon (which can be very severe) and skin thickening (scleroderma) are the two most common symptoms of SSc. Although both create unpleasant and frequently severe symptoms, however, it is the involvement of the internal organs that makes SSc life-threatening.

Briefly, SSc patients are divided into lcSSc or dcSSc based on clinical features, including skin involvement, Raynaud's phenomenon, musculoskeletal symptoms, calcinosis cutis, and characteristic autoantibodies (Herrick, 2018).

- **Skin involvement**: Skin thickening is a very common feature of SSc. For example, in dcSSc, increased skin involvement can lead to more severe internal organ failure. Increased skin thickening is due to increases in collagen. A modified Rodnan skin score (mRSS) is the most appropriate measure for skin disease. This method is calculated by summation of the skin thickness on 17 surfaces of the body. The score ranges from 0, which means no thickening, to 51 which means severe thickening (Bosoni, 2016).
- **Raynaud's phenomenon**: Raynaud's phenomenon is an early clinical feature and the most common, affecting 96% of SSc patients. Organ complications may not manifest until later in the course of the disease. Distressing physical symptoms, impaired function, body image dissatisfaction, and reduced quality of life are associated with Raynaud's phenomenon. Studies have reported cold fingers, colour changes in the skin, and numbness as common symptoms of Raynaud's phenomenon. The fingers and toes are the main body parts affected (Goundry et al., 2012).
- **Characteristic autoantibodies**: Autoantibodies are found in more than 95% of SSc patients. These include anticentromere antibodies (ACA), anti-topoisomerase (TOPO), anti-U1-RNP (U1-RNP), anti-RNA polymerase III (Pol 3), and anti-U3-RNP (U3-RNP). It has been found that these autoantibodies are associated with a specific demographic, affecting specific organs. Organ failure may be associated with a particular antibody; therefore, the identification of the antibody can lead to better diagnosis and treatment (Steen, 2005).
- **Musculoskeletal**: Musculoskeletal involvement is very frequent in SSc patients and causes arthralgia, synovitis, and contractures (Bosoni, 2016).
- **Calcinosis cutis**: Calcinosis cutis refers to indissoluble calcium in the skin and is a common symptom in SSc patients. It mainly affects the fingers, and there is no current, optimal treatment (Bosoni, 2016).

Table 3.1 summarizes the clinical features of the dcSSc and lcSSc subsets.

Table 3.1 *: Subsets of Systemic Scleroderma: Main Features of Limited Cutaneous Systemic Sclerosis Compared to Diffuse Cutaneous Systemic Sclerosis (Bosoni, 2016)*



Characteristic feature	Limited cutaneous SSc	Diffuse cutaneous SSc
Skin involvement	Indolent onset; slow progression; limited to fingers, face, distal to elbows	Rapid onset and progression; diffuse on fingers, extremities, face and trunk
Raynaud's phenomenon	Antedates skin involvement, sometimes by years; may be associated with critical ischemia in the digits	Onset coincident with skin involvement; critical ischemia less common
Muscoloskeletal	Mild artralgia	Severe arthralgia, carpal tunnel syndrome, tendon friction rubs
Calcinosis cutis	Frequent, prominent	Less common, mild
Characteristic autoantibodies	Anticentromere	Anti-topoisomerase I (Scl-70), anti-RNA polymerase III

3.2.3 Prognosis and Diagnosis of Systemic Sclerosis

The mortality rate of SSc is high and has not substantially changed in the last 40 years. A study in 2010 on 5860 SSc patients shows that SSc patients have a high risk of cancer. Additionally, the research indicates that 35% of SSc deaths were caused by lung fibrosis and 26% by heart failure. In SSc, some patients deteriorate quickly and may die, but others remain stable with few symptoms. The different clinical phenotypes and clinicians' limited knowledge to predict the risk of future organ-system complications and the prognosis show that the management of SSc is challenging (Tyndall et al., 2010).

The diagnosis of SSc is based on clinical assessment, and it has been suggested that the appearance of Raynaud's phenomenon followed by skin thickening and other extracutaneous features are the main symptoms of SSc. The diagnosis of this disease in the first months is quite difficult, as the only symptom is soft tissue swelling. However, some features, such as calcinosis and telangiectasia, can help physicians to make the appropriate diagnosis. Also, one or more of the following clinical features can help clinicians to confirm the disease (Hachulla & Launay, 2011):

- Heartburn
- Acute onset of hypertension and renal insufficiency

- Dyspnoea on exertion (associated with interstitial lung disease)
- Diarrhoea with malabsorption
- Facial, tongue, lip, or hand telangiectasia
- Digital ulcers or digital pitting scars or both
- Typical microvascular changes on nailfold capillaroscopy

Furthermore, the presence of below autoantibodies may be indicators of SSc (Kayser & Fritzler, 2015).

- **Anti-topoisomerase I antibodies (ATA)**: ATA, or anti-Scl-70, has been found in 15–42% of SSc patients with a specificity ranging from 90% to 100%. They are strongly linked with dcSSc and a poor prognosis. A higher risk of severe pulmonary fibrosis and cardiac involvement has been associated with SSc patients who have ATA. In addition, the presence of ATA with Raynaud's phenomenon in patients can indicate a high risk of developing SSc (Kayser & Fritzler, 2015).
- **Anti-centromere antibodies (ACA)**: ACA were first described in 1980. A study has described ACA as the most diagnostic antibody indicator of SSc, as it is commonly detected in SSc patients. These antibodies are associated with lcSSc. ACA is said to be associated with a higher risk of pulmonary arterial hypertension (PAH) and mortality (Kayser & Fritzler, 2015).
- **Anti-RNA polymerase antibodies (ARA)**: ARA were described in 1990 and are present in 5–31% of SSc patients. In common with ATA, they are associated with dcSSc. SSc patients with ARA have a high risk of developing renal crisis, joint contractures, and malignancies (Kayser & Fritzler, 2015).

3.2.4 Organs Involvement

SSc is not solely a skin disease but can affect multiple organ systems, including the lungs, kidneys, heart, and gastrointestinal tract. As mentioned previously, patients with dcSSc can progress more rapidly, and organ complications can occur earlier and may be worse than in lcSSc patients. Figure 3.3 illustrates the impact of SSc on the body.

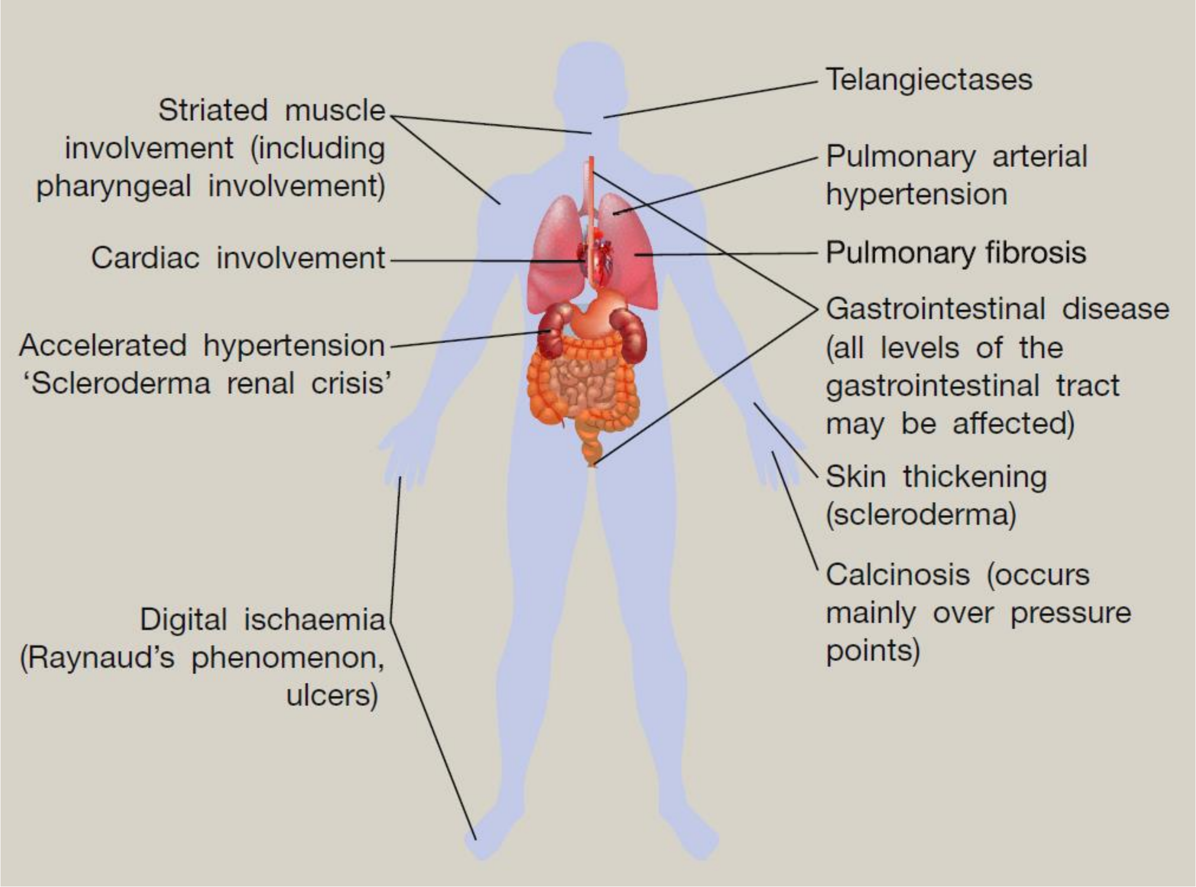

Striated muscle involvement (including pharyngeal involvement)
Cardiac involvement
Accelerated hypertension 'Scleroderma renal crisis'
Digital ischaemia (Raynaud's phenomenon, ulcers)
Telangiectases
Pulmonary arterial hypertension
Pulmonary fibrosis
Gastrointestinal disease (all levels of the gastrointestinal tract may be affected)
Skin thickening (scleroderma)
Calcinosis (occurs mainly over pressure points)

Figure 3.3: *Systemic Sclerosis – A Multisystem Disease (Herrick, 2018)*

- **Skin:** SSc may affect only the skin in the early stages of the disease and manifests as skin thickening and shiny areas around the mouth, bones, and fingers. The skin is the largest organ of the body and can be affected when disease spreads over the body. If the skin thickening and shiny areas are widespread, assessment of the disease may be challenging; assessment of other organs, such as the heart and kidneys, may be less difficult. Skin assessment is performed using skin scoring (Herrick, 2018).

- **Lungs:** The lungs of SSc patients are typically affected as may be observed using X-rays. Lung involvement is also the main cause of disability and death. The prominent lung complications in SSc are pulmonary arterial hypertension (PAH) and pulmonary fibrosis (PF). PAH is a serious complication of SSc that can affect both dcSSc and lcSSc patients. It is a progressive disease characterised by an increase in blood pressure in the arteries of the lungs. PAH can lead to heart failure and death. PF is another serious complication, which can affect 75% of SSc patients. It affects very small areas in the lung and disturbs pulmonary function when the forced vital capacity, the diffusing capacity for carbon monoxide, is less than 55% of normal (Yaghi et al., 2020).

- **Kidneys.** Kidney failure is a severe complication that appears in dcSSc patients who have had the disease for less than five years. It starts when blood flows to the kidneys, and this process is known as renal crisis. It is assessed by measuring blood levels of creatinine. Renal crisis affects 5–10% of SSc patients and is associated with high blood pressure, so patients are typically encouraged to monitor their blood pressure. If blood pressure is elevated above 160/90 twice in 12 hours, patients need further evaluation. Patients with renal crisis may present with headache and hypertensive retinopathy, associated with visual disturbances. Historically, mortality related to renal crisis has been high; however, deaths decreased from 42% to 6% between 1972 and 2002. Most SSc patients who develop renal crisis are dcSSc patients, which harmonises with a study from the United Kingdom that reports that 12% of dcSSc and 2% of lcSSc patients developed renal crisis (Bruni et al., 2018).

- **Heart.** The heart fails at least temporarily when the kidneys fail. In the same way, when kidney function is optimal, heart function can return to normal. Therefore, renal crisis and pulmonary hypertension can lead to significant cardiac issues, depending on the kidney and lung damage. Cardiac involvement can be classified as direct effects and indirect effects. Direct myocardial effects include cardiac failure and cardiac fibrosis; indirect myocardial effects include other organ involvements. The symptoms of cardiac involvement in SSc are varied, but shortness of breath with tiredness and paroxysms are the main symptoms. It has been shown that cardiac involvement is more frequent with dcSSc (Champion, 2008).
- **Gastrointestinal Tract.** A substantial proportion of SSc patients (80–90%) have lazy muscles in their oesophagi. This can cause heartburn as food sticks in the chest. In addition, stomach muscle can be lazy, and this can create a feeling of fullness after minimal ingestion of food. A progressive gastrointestinal process is involved from grade zero vascular damage to grade one neurogenic impairment and then grade two myogenic dysfunction. The gastrointestinal region is considered the second site of damage by SSc (Forbes, 2009).

3.3 Data Collection

The dataset utilised in this research was obtained from the Royal London Hospital. Its subject is SSc disease, and it was collected from 677 patients. The features of the dataset are considered in the following sections.

3.3.1 General and Subset Data

- Subset: This refers to the subgroup of SSc and includes two values. First is the limited cutaneous subset (L), which includes individuals who do not have skin thickening near their elbows and knees; this group is labelled ‘1’ in the dataset. The second is the diffuse cutaneous subgroup (D), which includes patients with skin thickening in both the distal and proximal areas of the elbows and knees; this group is labelled ‘2’.
- Gender: This variable indicates the patient’s gender: ‘m’ for males and ‘f’ for females.

- Age: The age in years of a patient at disease onset (integer values).

3.3.2 Blood Tests Results

- abs: The autoantibodies that have been found. The dataset has 16 columns, each of which is headed by a specific autoantibody acronym. Each column includes a binary value to indicate the autoantibody's absence or presence: '0' indicates absence and '1' indicates presence (binary values).
- Hb: Haemoglobin concentration in grams per decilitre. The typical range for men is 13.5 to 17.5 grams per decilitre, and the usual range for women is 12.0 to 15.5 grams per decilitre.
- Cr: Creatinine, an indicator of the stage of kidney disease. It can be calculated by serum creatinine level, age, sex, and race. The baseline for Cr is between 60 and 90 ml/min/1.73m^2.

3.3.3 Lung Function Test Results

- FVC: Forced vital capacity, measured in litres.
- DLCO: Carbon monoxide diffusing capacity, expressed in litres.
- T2RIP: The number of months between the development of the sickness and mortality.
- T2PF: The number of months between the commencement of the disease and the development of pulmonary fibrosis.
- T2PAH: The number of months between the commencement of the disease and the onset of pulmonary arterial hypertension.

3.3.4 Antibody Information

The following antibodies are marked in the dataset with binary values ('1' or '0'):

- ACA is the most common and is linked to the lcSSc subgroup. A small number of ACA positive people can develop dcSSc.
- ATA is linked to an increased risk of arthritis, tendon friction rubs, severe pulmonary fibrosis, heart involvement, and scleroderma renal crisis.
- ARA is strongly associated with the dcSSc subset and correlated with severity of skin involvement

3.4 Machine Learning Methods

ML is commonly used in the field of medicine to assist the decision-making of physicians. In general, ML uses two main techniques: supervised learning and unsupervised learning. Supervised learning methods, such as DT, random forest, and NB, train a model using known inputs and outputs of patients to predict the unknown outputs of new patient details. Unsupervised learning methods, such as K-means and LCA, find the relationship between cases by discovering hidden patterns between input data without labelled responses. ML methods are becoming useful tools in disease diagnosis, drug development, complication prediction, and personalized treatment. However, the accuracy of results can vary depending on the method, hence selecting an appropriate method is important. The various ML algorithms that have been implemented on the dataset are described in the following sections.

3.4.1 Decision Tree (DT)

In medicine, DT algorithm is the most widely used supervised-learning approach. It is a decision support model that is structured using a follow-up chart from root to leaf. It creates a classification model by reviewing dataset observations to predict class labels (Podgorelec et al., 2002).

A DT structure has the following elements:

1- **Root node:** The root node is the first node of the tree and has no incoming edges. This root node is divided into sub-nodes and represents the best attribute of the dataset.

2- **Internal nodes:** This node has one incoming edge and two or more outgoing edges. It represents an attribute.

3- **Leaf nodes:** This node has no outgoing edges and only one incoming edge. It represents a class label.

4- **Branches:** The branch indicates action, so each branch represents the outcome value of the node.

The following Figure 3.4 shows a DT structure using a general example

7

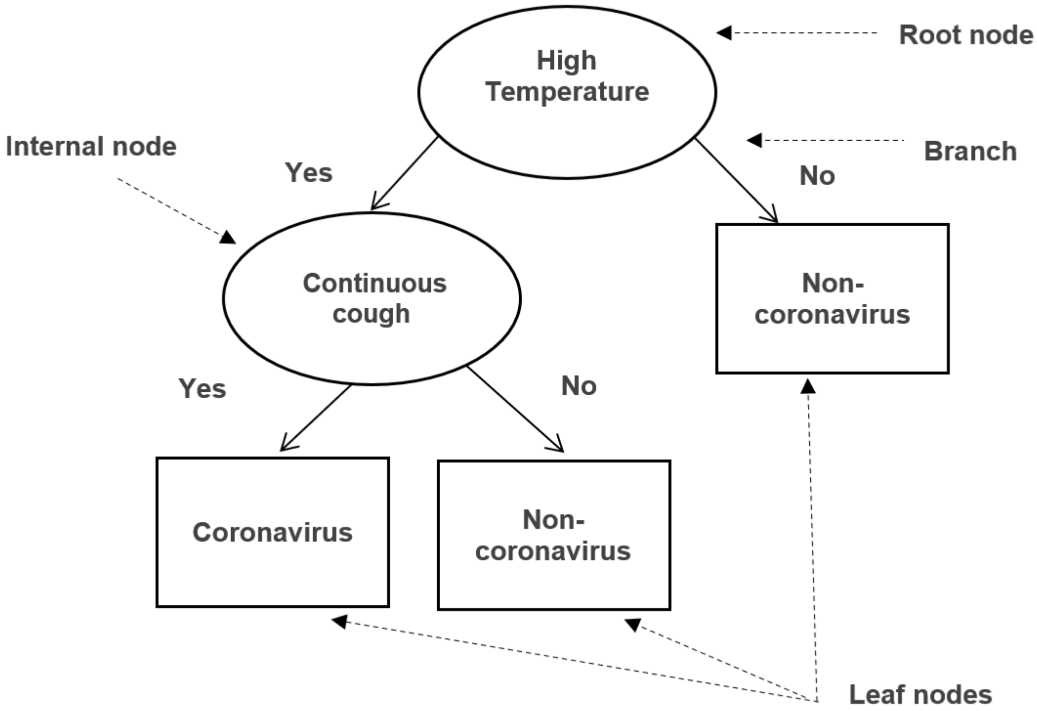


Figure 3.4: *A Decision Tree for Diagnosing Coronavirus Disease*

A DT can be a powerful method for making predictions in medicine. It aims to find the optimal DT by minimizing the generalization error for a given dataset. Finding the optimal DT can be difficult; therefore, heuristic methods may be used. C4.5 is a method that create DTs using a pruning phase. Pruning is an important phase for complex problems, as it removes nodes that do not provide additional information and improves accuracy.

The following are the steps for constructing a DT model using a given dataset.

1- Identify the dataset classes.
2- Find the optimal attribute, which will become the tree's root node.
3- Subdivide the training dataset based on the optimal attribute values (branch values). For instance, according to Figure 3.4, the best attribute is high

temperature, so the training dataset is divided into patients with high temperature and patients without high temperature.

4- Calculate the ratio of the information gain metric for each attribute of the new subset.

5- For this subgroup, choose the attribute with the highest value of information gain ratio as the best attribute. This attribute is called an internal node and is used to again divide the training dataset into subsets.

6- Repeat steps 2 to 5 until the optimal information gain is 0, signifying that the node is a leaf (a class label).

Information gain is computed using the following formula:

$$\textit{Information gain}\ (\mathrm{S},\mathrm{A}) = \textit{Entropy}(\mathrm{S}) - \sum_{v\epsilon V\ values(\mathrm{A})} \frac{|S_v|}{|\mathrm{S}|}\ \textit{Entropy}\ (S_v). \qquad (3.1)$$

$$\textit{Entropy} = -\sum_i\ p_i .\ log_2\, p_i \qquad (3.2)$$

V: all potential values for attribute A.

S_v: the subset of S.

p_i: the proportions of each label's elements in the set.

Algorithm 3.1 C4.5 (D)

```
Input: Dataset D
Output: Decision Tree
Classes = C
Attributes = A
For all attribute a ϵ D do
    Compute information gain
End for
$a_{optimal}$= best attribute
$a_{optimal}$ is the root of the Tree
d = sub-dataset from D based on $a_{optimal}$ values
For all attribute d do
    Compute information gain for all attributes ϵ d
    If information gain is zero then
```

```
12: Return leaf node (class label)
13: End if
14: Tree_d=C4.5(d)
14: Assign Tree_d to the appropriate branch of the Tree.
15: End for
16: Repeat 8 to 15 until all information gain is zero.
17: Return Tree
18: End
```

I applied C4.5 to my SSc dataset. All outcomes and assessments are detailed in the next chapter.

3.4.2 Random Forests (RF)

The RF algorithm is a collection of multiple DTs that are modelled for a prediction and analysis task. It can be applied to a categorical response variable, a known classification task, or to continuous response variables in a regression task. Thus, RF algorithms can be applied to classification and regression problems, and they are also fast algorithms that can be used to train a model and predict labels. In addition, RF algorithms can be applied to high-dimensional problems, and they are considered straightforward algorithms. Nevertheless, RF can compute the importance of variables when a model is created: this helps researchers to understand and interrupt results (Cutler et al., 2012).

The following are the steps for constructing a RF classification model for a given dataset.

1- Identify the dataset classes.
2- Divide the dataset into a training dataset and a test dataset.
3- Divide the training dataset into many subsets using the bootstrap approach.
4- Build a DT for each subset according to the instructions in the prior step.
5- Predict each case in the test data, based on each DT.
6- Make the final prediction for each case, which will be the voting class of all DT predictions.

The following figure shows an example of a RF classification algorithm.

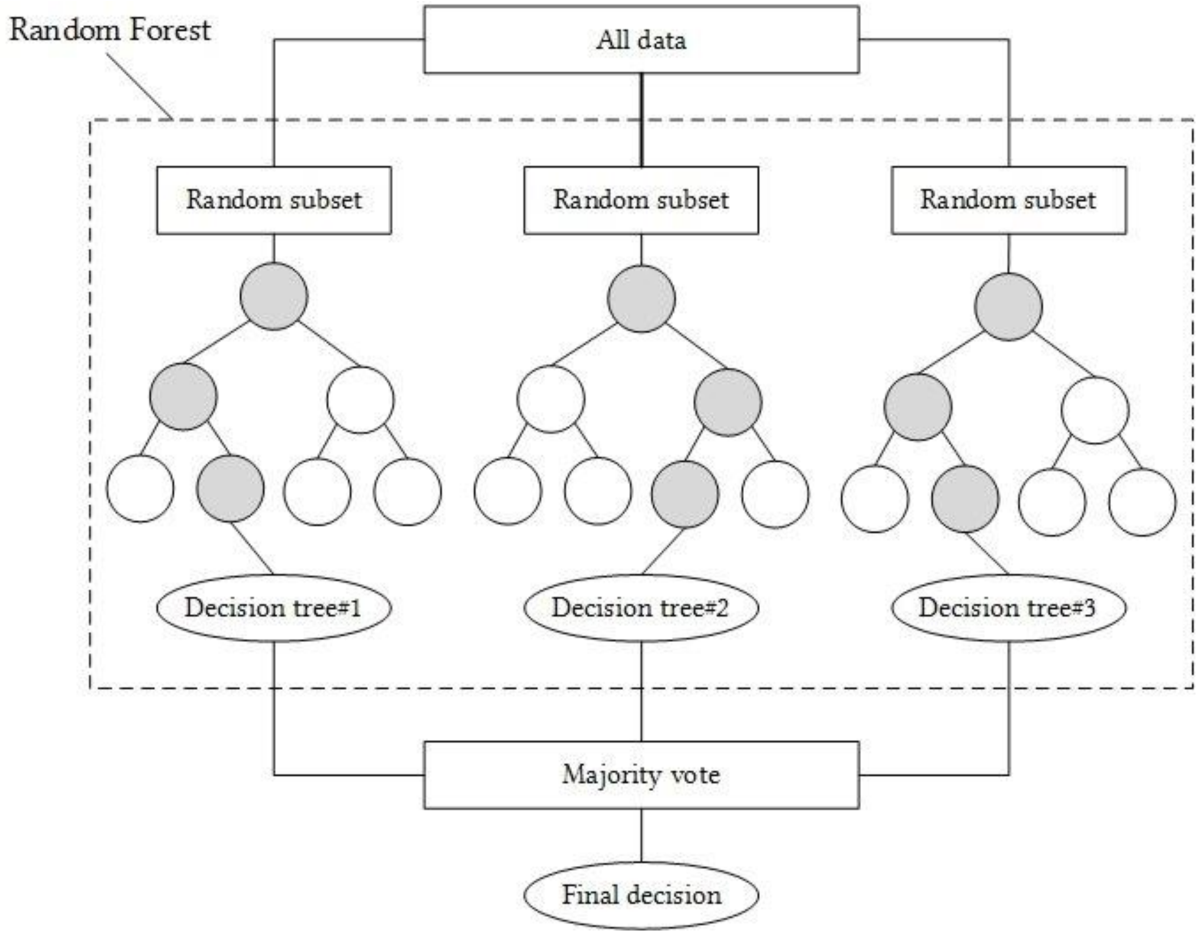


Figure 3.5*: A Simple Example of a Random Forest Classification (Chapron et al., 2018)*

Algorithm 3.2 (RF)

Input: Dataset

Output: Voting prediction

```
Classes = C
Split the dataset into training dataset (D) and test dataset (T)
Assign M (the number of trees)
For 1 to M
    Create subset ($d_m$) of the training data using bootstrap random with replacement for training dataset
    Build a decision tree ($dt_m$) for ($d_m$)
End for
For 1 to test dataset ($t_i$)
    For 1 to decision tree ($dt_m$)
        Predict ($t_i$) class from ($dt_m$)
        Assign the class to ($dtt_{i\,m}$)
    End for
    Assign the optimal class for $t_i$ using the voting approach on ($dtt_{i\,m}$)
End for
End
```

Due to its benefits, the RF classification method was applied to both the SSc data and the synthetic COVID-19 data in my study. All outcomes and evaluations are described in Chapter 6.

3.4.3 Naïve Bayes (NB)

As with DT algorithms, NB algorithms are supervised ML methods that provide transparent explanations. An NB classifier is a probabilistic model based on applying the Bayes theorem. It considers the independence assumption between the attributes and calculates the probability of a hypothesis, given prior knowledge (Borkar & Deshmukh, 2015). For example, if I want to predict whether a patient will develop PAH within five years or after five years using my SSc dataset, the NB classifier will provide the probability.

Bayesian classifiers employ the Bayes theorem, which states the following:

$$p(c_j \backslash d) = \frac{p(d \backslash c_j) P(c_j)}{p(d)} \qquad (3.3)$$

$P(c_j \backslash d)$: the probability of instance d being in class c_j

$p(d \backslash c_j)$: the probability of generating instance d given class cj

$p(d)$: the probability of instance d occurring

$P(c_j)$: the probability of class cj occurring

Assume there are two classes: C1 = 1 and C2 = 2

1: An SSc patient may develop PAH within five years.

2: An SSc patient may develop PAH after five years.

Suppose I have a patient, X, who has SSc; according to the preceding equation, the likelihood of X developing C1 = 1 is the probability of X patient given that he develops C1 = 1 multiplied by the probability of being C1 = 1 and divided by the probability of being patient.

As mentioned in the preceding chapter, an NB classifier may be used effectively in medicine, as it is fast and not sensitive. In addition, it can handle real and discrete data. However, NB assumes the independence of features, and this is a disadvantage. In my research, I applied a standard NB classifier to my SSc dataset. All outcomes and assessments are detailed in Chapter 5.

3.4.4 K-Means

Clustering methods are an unsupervised ML technique for grouping data without prior knowledge of group definitions; clustering algorithms are also used to detect natural groupings in unlabelled data. K-means is a clustering technique that splits the dataset (patients) into distinct groups (clusters) in which all members share comparable features. In medicine, K-means can be used to identify subgroups and subclasses of disease.

The K-means algorithm is a simple technique. It begins with selecting the desired number of clusters (K). K starting centroids are determined randomly, and each point in the dataset is allocated to the centroid that is the closest. Thus, all points allocated to a centroid constitute a cluster. The second stage modifies the cluster centroids based on the points assigned to each cluster, and again each point in the dataset is allocated to the centroid that is the closest. This stage is repeated until the centroids remain the same. The output is K clusters (groups) with distinct characteristics (Nithya et al., 2014).

The following demonstrates the K-means pseudocode required to execute the K-means algorithm on a dataset.

Algorithm 3.3 (K-means)

```
Input: dataset D
Output: k clusters (groups)
Select randomly K points as initial centroids
Repeat
Assign each point to the closest centroid
Recompute the centroid of each cluster
Until centroids do not change
Return K clusters
```

To assign each point to the centroid closest to it, I require measurements to compute the closest centroid to a point in Euclidean space. For Euclidean data, both Euclidean distance and Manhattan distance can be used.

- **Euclidean Distance**

This formula computes the distance between two points, x and y. It is a metric to measure the line between two points. The equation for this metric is as follows:

$$\mathbf{Dist_{xy}} = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \quad (3.4)$$

- **Manhattan Distance**

Manhattan distance is also called ‘city block’. It measures the distance between two points by the absolute difference between the points. The equation for this metric is as follows:

$$Dist_{xy} = \sum_{i=1}^{n} | x_i - y_i | \quad (3.5)$$

The K-means algorithm was applied to my SSc dataset, and the results are provided in the following chapter.

3.4.5 Latent Class Analysis (LCA)

LCA is a statistical method that discovers subgroups of patients who share common characteristics. It is a probabilistic approach that finds the most likely model. The model resembles the clustering method in that patients are grouped together. In addition, it applies a probabilistic methodology similar to the NB method to identify the most probable model. LCA finds the hidden relationships between the observed variables to cluster the data within groups (clusters). The number of subgroups is selected by running a number of analyses starting from a model with one group and adding more subgroups until the optimal solution is found: the optimal solution is the one that clusters the dataset well in groups.

This model helps clinics to find groups of patients with similar demographics, clinical characteristics, treatments, comorbidities, and outcomes. LCA has therefore become a very popular method in the healthcare industry (Mori et al., 2020). In my research, SSc patients can be divided into groups using this method by finding the relationships (hidden factors) between the observed variables. Thus, the latent (hidden) variables can be expressed as follows:

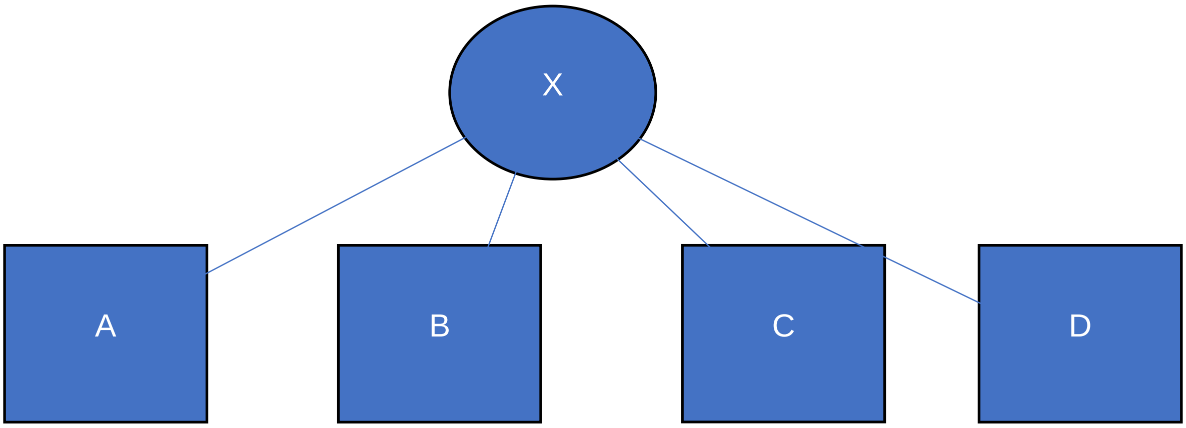


Figure 3.6: *Latent Class Analysis Structure. X Is the Latent Categorical Variable. A, B, C, and D Are Observed Variables*

I implemented this algorithm on my SSc dataset: all results and implementation details are described in Chapter 5.

3.4.6 Drift Detection Method (DDM)

This method monitors a model's performance over time. It detects a drift when the model becomes outdated and updates the model accordingly. Gama et al. (2014) created this method, which analyses the total classification model's error rate. This method assumes that the overall error rate of a classifier will either decrease or remain constant when the number of samples increases. In the equations below, assume p_i is the error rate of the classifier, s_i is the standard deviation of the classifier of sample *i,* p_{min} is the minimum error rate recorded of the classifier, and s_{min} is its standard deviation (Jaramillo-Valbuena et al., 2017). This algorithm functions as follows:

- $p_i + s_i > p_{min} + s_{min}$

The model is operating normally with no alarms.

- $p_i + s_i \geq p_{min} + 2 * s_{min}$

Future drift is possible, and the model is now in the warning zone.

- $p_i + s_i \geq p_{min} + 3 * s_{min}$

There is drift, and the model needs to be updated.

I implemented this algorithm on the SSc and syntactic COVID-19 datasets. All results and implementation details are described in Chapter 6.

3.5 Resampling Methods

ML models use resampling techniques to increase the model's performance and validate the model. Cross validation and bootstrapping are the two most important strategies.

3.5.1 Cross Validation

Cross validation is a statistical technique for comparing and evaluating learning algorithms. Cross validation is a re-sampling method that draws samples from a dataset and fits each sample to a model to obtain addition information about this model. It works by separating the dataset into two parts: the first contains the training data used to train a model, and the second contains the validation data used to validate the model. Both components must cross over in successive rounds for each data point to be validated.

The most common type of cross validation is K-fold, in which the data is divided into k subsets evenly, and each subset is called a fold. Subsequently, for each K, a dataset is withheld for validation in each iteration of a separate fold, and the K-1 folds are used as the learning dataset (Refaeilzadeh et al., 2009).

Figure 3.7 illustrates a 10-fold process of cross validation. Thus, the dataset is separated into 10 groups, with each group containing a unique test dataset. Therefore, the validation data is 1/10 of the total training dataset. K = 10 is considered a good choice, as it allows overlapping in the training set and keeps the test set independent.

Leave-One-Out cross validation (LOOCV) is a special case of K-fold cross validation where each observation can be used for the validation set while the other points become the training set. Although LOOCV has less bias, it is very expensive to implement because the model has to fit the number of observations (Berrar, 2018).

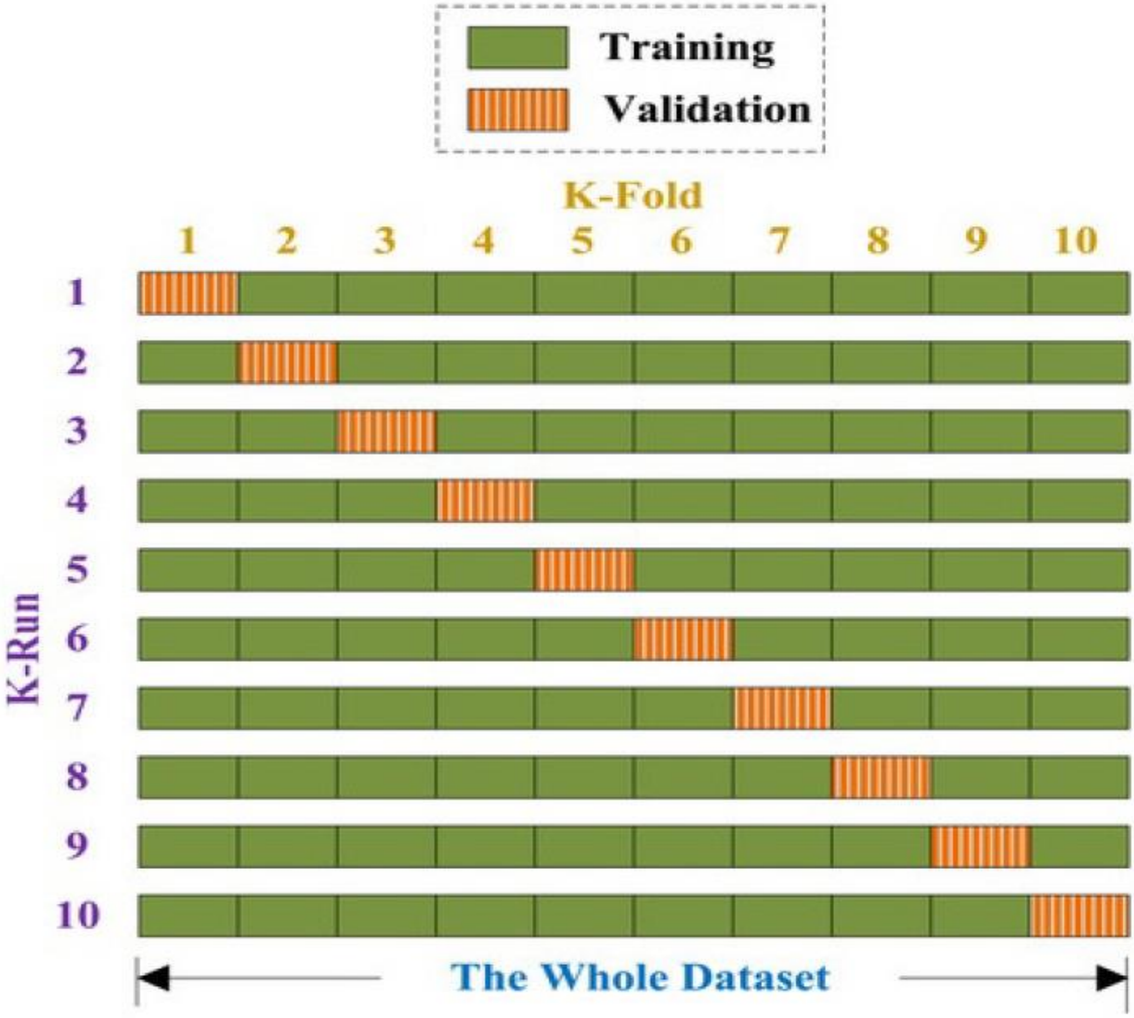


Figure 3.7: *Diagram of a 10-Fold Cross Validation (Bosoni, 2016)*

Algorithm 3.3 (K-Fold Cross Validation)

```
Input: Dataset D
1: Divide the original dataset into k groups
2: For each resampling iteration do
3: Hold out specified sample as validation set
4: The mean squared error is computed on the data points in holdout sample
4: Fit the model in k-1
5: Predict the hold out sample
6: End for
7: Calculate the average performance across all predictions
8: End
```

Mean squared error is computed as follows:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \quad (3.6)$$

n: the number of observations

y_i: observed values

$\hat{y}_i$: predicted values

A K-fold cross validation estimate is computed as follows:

$$\text{CV}(n) = \frac{1}{n}\sum_{i=1}^{n} MSE_i \qquad (3.7)$$

3.5.2 Bootstrapping

It has been proven that resampling methods, such as cross validation and bootstrapping, are more accurate than classical methods. In addition, these techniques permit the quantification of uncertainty by calculating standard errors and confidence intervals. The bootstrapping method creates hundreds or thousands of additional samples that are drawn from the original data. This is achieved by taking replacement samples (resamples) from the original sample. The size of each resample is the same as the original sample. This distribution of resampled statistics is known as the bootstrap distribution (Pottel & Hans, 2015).

The following steps, pseudocode, and Figure 3.8 describe the bootstrapping process:

1- Determine the number of samples to be drawn from the original dataset.
2- Select observations at random from the original dataset. Each data point may appear many times per sample.
3- Sample with replacement. The sampled data is returned to the original dataset so that it can be used for the subsequent sampling.
4- Compute the bootstrap distribution by calculating the statistics for each sample and the distribution of resamples.
5- Obtain information about the population from the bootstrap distribution.

Algorithm 3.4 Bootstrapping

```
Input: Dataset D
Select the number of samples i
For each sample do
    Randomly select observations from D with replacements
    Compute the sample mean
End for
Compute the bootstrap standard error
End
```

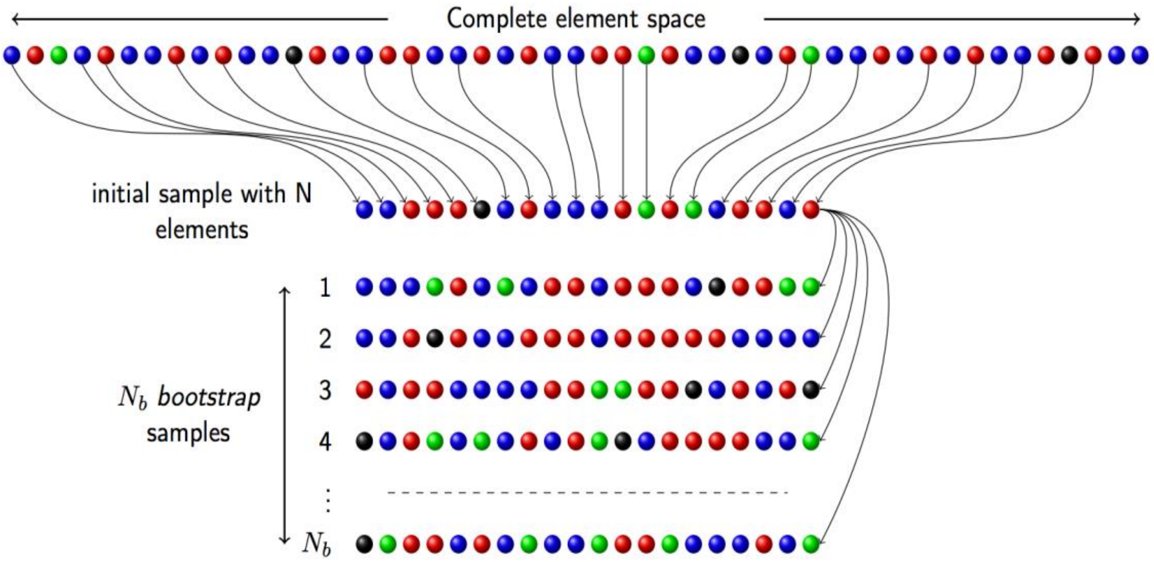


Figure 3.8: Diagram of a Bootstrapping Resampling Method

To summarise, both cross validation and bootstrapping can be used to evaluate models. Both are simple to implement and are widely used approaches. To test the new models, I incorporated these resampling approaches in my new algorithms.

3.6 Evaluation

When a ML model is constructed, its performance must be evaluated. A superior performance model results in a superior model. Unsupervised and supervised learning models are often evaluated with a variety of metrics.

3.6.1 Unsupervised Learning Measures

In clustering, individuals assigned to a particular group should be in close proximity (compactness), and the groups should be spaced out (separation). Therefore, an effective clustering algorithm means members in each group are highly similar. The features of the dataset and their values can play an important role in the performance of a clustering algorithm. Hence, it is important to evaluate the validity of the model through cluster validation. There are four approaches to investigate cluster validity: external criteria, internal criteria, relative criteria, and stability criteria (Halkidi et al., 2001).

- **External indexes**

This method compares the clustering algorithm results with externally provided, known results. This method can be divided into three categories: pair counting, information theoretic, and set matching. Pair counting measures include rand index and adjusted rand index that count pairs of objects in the dataset in two different clusters and determine if they agree or disagree. Information theatrics, such as entropy, measure the information that two clusters share. Set matching, such as F measures, is based on pairing similar clusters. External indexes have been used in genetic algorithms to measure genetic diversity in a population (Halkidi et al., 2001)

- **Internal indexes**

Internal indexes measure the goodness of the clustering algorithm without using external information. Internal index measures can find the optimal clustering algorithm and the optimal cluster number. They focus on compactness, which measures an object's relationship to others in the same cluster based on variance: lower variance indicates better compactness. In addition, it focusses on measures that provide information on the degree of separation between groups. Measuring the silhouette coefficient is a method that combines compactness and separation (Halkidi et al., 2001).

- **Relative indexes**

This method is used to compare two different clusters. It attempts to measure the consistency of an algorithm by using the same algorithm under different conditions. Relative indexes measure the stability of an algorithm against a different dataset. A weighted-kappa coefficient, which measures the degree of disagreement between two categories, can be used to measure the stability of a clustering algorithm. This statistic measures the agreement among the decisions made by two or more observers and returns a score between zero and one. Zero means the agreement is poor; one means perfect agreement (Bosoni, 2016).

The formula for weighted kappa (WK) can be expressed in the following equation:

$$K= 1 - \frac{\sum_{i=1}^{k}\sum_{j=1}^{k} w_{ij}x_{ij}}{\sum_{i=1}^{k}\sum_{j=1}^{k} w_{ij}m_{ij}} \qquad (3.8)$$

K: the number of observers

w_{ij} , x_{ij}, m_{ij}: the elements in the weights

Table 3.2 Weighted-Kappa Guidelines

Weighted kappa	Agreement strength
K < 0.0	Less than random
0.0 < K < 0.2	Poor
0.2 < K < 0.4	Fair
0.4 < K < 0.6	Moderate
0.6 < K < 0.8	Good
0.8 < K < 1.0	Very good

3.6.2 Supervised Learning Measures

A variety of performance metrics can be used to evaluate a supervised learning algorithm. These include accuracy, sensitivity, specificity, and precision, which are typically measured performance characteristics that can be determined from a confusion matrix. Classification performance is best described by applying a confusion matrix, which is a binary contingency table that is used to describe the performance of a classification model. Table 3.3 shows the confusion matrix used in this research. Each row of the matrix represents the cases or patients in a predicted class and each column represents the cases in an actual class (Bahl, 2017).

Table 3.3 *Confusion Matrix*

		Prediction	
		Positive	Negative
Actual	Positive	TP	FN
	Negative	FP	TN

TP: True positive – a correct positive prediction.

TN: True negative – a correct negative prediction.

FN: False negative – an incorrect negative prediction.

FP: False positive – an incorrect positive prediction.

The ‘error rate’ (ERR) is the total erroneous data (incorrect predictions) compared to the total data (all patients), and the ‘accuracy’ (ACC) is the total number of correct data predictions compared to the total data. These can be derived from the confusion matrix as follows:

$$\text{ERR} = \frac{FP + FN}{TP + FN + FP + TN} \qquad (3.9)$$

$$\text{ACC} = \frac{TP + TN}{TP + FN + FP + TN} \qquad (3.10)$$

Sensitivity (SN) and specificity (SP) can also be derived from a confusion matrix. ‘Sensitivity’ is the ratio of instances that have been predicted correctly and positively (TP) to the total number of actual positives in the data (TP and FN). In other words, it means the proportion of TPs that are correctly identified.

$$\text{SN} = \frac{TP}{TP + FN} \qquad (3.11)$$

‘Specificity’ is the ratio of instances that have been predicted correctly and negatively to the total number of actual negatives in the data. In other words, it means the proportion of actual negatives that were correctly predicted.

$$\text{SP} = \frac{TN}{TN + FP} \qquad (3.12)$$

‘Precision’, or positive predictive value (PPV), is the ratio of instances that have been predicted positively and correctly to the number of instances that are TPs and FPs.

$$\text{PPV}= \frac{TP}{TP + FP} \qquad (3.13)$$

When a model needs to be examined, the above metrics may be employed. They are used to evaluate my new models in the next chapters.

Finally, a receiver operating characteristics (ROC) curve can be used to select an optimal model. In addition, the ROC curve explains the performance of a binary classifier by plotting the TP rate (sensitivity) against the FP rate.

3.7 Summary

SSc is a rare disease that affects the skin and other organs, and it remains difficult to diagnose due to a lack of information and a limited number of patients. ML algorithms such DT, RF, NB, K-means, or LCA can be useful tools to assist the decision-making of physicians. This chapter presented SSc disease as a rare disease and the complications of this disease. Numerous ML algorithms were described that aid in the analysis of SSc. The chapter also provided a DDM method for monitoring ML models over time. In addition, it offered resampling techniques, such as cross validation and bootstrapping, that have the potential to improve the performance of ML models. This chapter concluded by presenting the evaluation metrics used to evaluate ML models.

In the next chapter, I present the new model, the ‘Nearest Consensus Clustering Classification’ and use it to identify subclasses and predict SSc.

Chapter 4
Nearest Consensus Clustering Classification to Identify Subclasses and Predict Disease

4.1 Introduction

Disease subtyping, which helps to develop personalised treatments, remains a challenge in the data analysis field because there are many different ways to group patients based on their data. However, identifying disease subclasses would enable the development of better, more personalised models; this would thereby improve prediction and understanding of the disease's underlying characteristics. This chapter provides a new method that combines consensus clustering techniques with classification methods to improve disease prediction and awareness of underlying disease characteristics. As such, this chapter first investigates the significance of disease subtyping and its influence on the performance of ML models. It describes consensus clustering and the procedure for implementing consensus clustering methods. Second, this chapter describes the proposed consensus clustering method employed in my research to predict disease as well as the datasets I used and the experiments I conducted. Finally, my experiment, in which the findings of the proposed method were applied to a real-world freely available breast cancer dataset from a London hospital on SSc, is described and interpreted. This chapter was published as an article in *Journal of Healthcare Informatics Research* in 2018.

4.2 Background

Disease subtyping aids in the development of personalised medicines, which are better suited to specific patients. However, barriers to data analysis remain due to the numerous techniques that can be used to cluster patients based on their data. However, if I can identify disease subclasses, this would aid the construction of better, more specialised models for certain patient groups, thereby enhancing my ability to anticipate and comprehend the underlying characteristics of a given disease. Cluster techniques have been proven effective in this field. In medicine, clustering approaches, which are often used to separate thousands of patients into manageable groupings, can provide numerous benefits (Kellam et al., 2001). However, traditional algorithms such as K-means, DB-scan, and fuzzy c-means might be biased and of inconsistent quality due to limited sample sizes, intrinsic model bias and noise. Consequently, consensus clustering strategies have been developed (Kalyani, 2012).These approaches have traditionally addressed model bias and variability but not sample variance, which is addressed in this chapter using resampling techniques.

The importance of discovering subtypes has increased as more data has become available. Clear cell renal cell carcinoma (ccRCC) is one of the most significant subtypes of renal cell carcinoma, according to Wu et al. Their study emphasised the significance of molecular typing for both individualised cancer treatment and the enhancement of overall accuracy. Unsupervised consensus clustering was utilised to identify a new subgroup of ccRCC. An unsupervised consensus clustering approach enabled the identification of three distinct subtypes based on hierarchical clustering. This is essential due to the capacity to define stable groupings based on patterns of gene expression. In addition, the clusters have clinical significance that may shed light on the behaviour and prognosis of the tumour (Wu et al., 2018). Zhu et al. proposed a novel subspace clustering guided unsupervised feature selection (SCUFS) algorithm that learns through representation-based subspace clustering. This method exposes the underlying multi-subspace structure of the data as it learns the data distribution. Results revealed that the SCUFS model outperformed alternative approaches (Zhu et al., 2017).

Choosing the appropriate clustering method is a complicated undertaking, as different methods might produce varying outcomes. Combining the results of many approaches can lead to better grouping. In addition, the bootstrap method (see chapter 3) can be used to resample datasets to increase confidence in clusters (Tucker & Garway, 2010). Consensus clustering, which investigates the consensus across many clustering algorithms, can boost overall confidence compared to each specific input cluster method. Even higher confidence can be given to robust clusters, which enforce maximum agreement across input clustering methods. Swift et al. used robust and consensus clustering in order to improve confidence in discovered clusters (Swift et al., 2004). Nguyen and Caruana presented a good review of consensus clustering methods (Nguyen & Caruana, 2007). The weighted-kappa metric (see chapter 3), which is used to assess the degree of concordance among the decisions of two or more observers, can be used to assess the consistency of clustering results. It can thus be used to compare different data allocations to clusters, generating a value that ranges from -1 to +1, indicating poor to extremely excellent agreement strength (Swift et al., 2004).

Once patient subclasses have been established, supervised learning can be applied to disease prediction. Decision trees and Bayesian classifiers perform well and have the added benefit of modelling data transparently, unlike many black-box approaches (Soni & Ansari, 2011). Tucker et al. developed a model integrating unsupervised and supervised learning to predict patient health outcomes. Their findings revealed that the model both enhanced physician understanding and improved prediction. I expand on their research by exploring how consensus methods can be used to identify individual models for each discovered subgroup, which aids understanding as well as improving prediction.

To this end, I analysed patients with SSc, and I describe this process in this chapter. Additionally, I integrated unsupervised learning, which discovers potential subclasses, with supervised learning, which helps predict health outcomes based on these subclasses. I designed a novel algorithm that performs better than supervised learning alone by incorporating unsupervised learning (K-means clustering). I named this algorithm ‘nearest consensus clustering classification identify subclasses and predict disease’. In the following subsection, I define consensus clustering before describing my novel method.

4.3 Consensus clustering

Multiple cluster results are combined in consensus clustering, which uses a variety of clustering methods as inputs to find a single consensus clustering method that is a better fit than any individual clustering method. Consensus clustering is required because it allows for the reconciliation of clustering data obtained from various experimental sources or repeated runs of the same non-deterministic algorithm (Goder & Filkov, 2008). It is also a method for finding clusters that are more stable and less sensitive to starting values based on a membership principle. It takes several input clustering methods into account, with items that have been grouped together repeatedly in the inputs having a higher chance of appearing in the consensus clustering. For example, to remove bias, consensus clustering might use many clustering methods, which have been formed using various clustering methods or starting parameters, as inputs (Xiao & Pan, 2007). Input clustering methods can also be developed by resampling the original dataset to remove sampling bias and generate a more stable consensus grouping.

The first step to implementing a consensus clustering approach is to create an $n \times n$ agreement matrix based on the input clustering results. This matrix comprises cells that show the number of agreements between the input clustering algorithms used to cluster each pair of objects, as indicated by the indexing row and column. This matrix is then used to sort items based on their cluster agreement by rewarding clusters with high member agreement and penalising clusters with low member agreement (Swift et al., 2004).

The input methods used to construct the agreement matrix can be the outcomes of several clustering techniques. In this case, however, I aimed to reduce sampling bias; therefore, I employed distinct clustering findings from the application of K-means clustering to numerous resamplings of the data. Consensus clusters that rewarded variables with high cluster agreement and penalised those with poor agreement were thereby created. Fig 4.1 provides an overview of how consensus clustering operates.

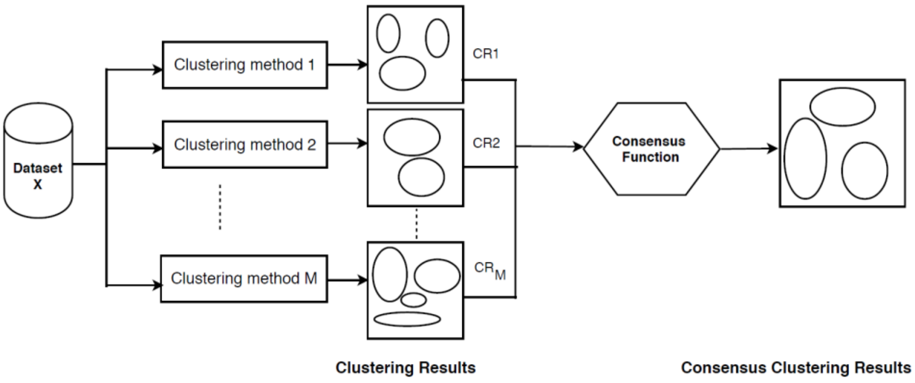


***Figure 4.1**: Consensus Clustering Algorithm (Schematic)*

4.4 Nearest consensus clustering algorithm

My proposed method attempts to account for the natural variation in many clustering methods as well as sample variance by using the consensus approach in combination with C4.5 DT classifiers. C4.5 is a transparent DT method for classification that provides a tree structure that can be comprehended. The information gain ratio measure is used to infer the tree (Balagatabi, 2013). As a result, my proposed method divides the data into two sets: the training set and the test set. To create a set of consensus clusters, the training data was resampled. Each of these consensus clusters was then used to generate a different DT. Then, using a single linkage approach with Euclidean distance (see chapter 3), each test data point was scored based on its distance from each detected consensus cluster. This was done to determine which DT should be used to classify the data point. Using this method, I examined a variety of distance measurements, such as single linkage (the closest element to point *a*), further linkage (the furthest element to point *a*) and average linkage (the average distance between point *a* and set of points). The suggested nearest consensus clustering algorithm is displayed in Fig 4.2 as a schematic diagram. In this example, the training data was divided into three clusters using consensus clustering of multiple K-means with resampled data. A DT was then constructed from each consensus cluster. When classifying test data, my algorithm aligned the test data (denoted by an ‘x’) to the nearest consensus cluster (here, cluster 3) using the single linkage measure (nearest neighbour). The associated DT was then used to classify the test data point (here, DT3).

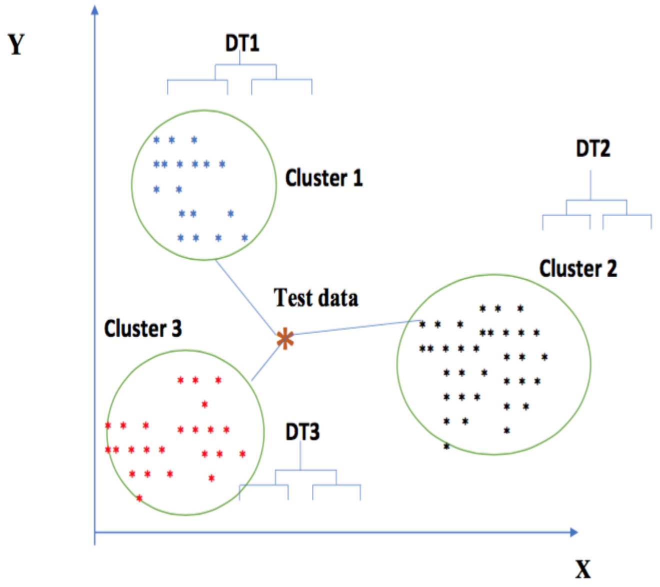


Figure 4.2: *Nearest Consensus Clustering Classification: Training and Testing Data (Schematic Figure)*

The following pseudocode explains the steps used to build the new algorithm.

Algorithm 1 Pseudocode of Nearest Consensus Clustering Classification

```
Input: Dataset of patients.
Output: Different clusters of patients and different DTs for each group.
Begin
1: For i = 1 to 10
    2: Randomly generate 80% training dataset and 20% test dataset.
    3: For k = 1 to 10
        3: Resampling with replacement the training dataset.
        4: Run K-Means on training dataset and store in InputClusters.
    5: End For
    6: Compute agreement matrix $(n \times n)$ $A$ from InputClusters
    7: Run hierarchical clustering on $A$ to generate consensus clusters (CC).
    8: Print CC (patient groups).
    9: Build decision tree (DT) for each group in CCs generated in step 6.
    10: For $j$ = 1 to the size of test dataset
        11: Compute Euclidean metric for test dataset patient ($j$) to each group in CC.
        12: Return the group that has the minimum value (mingroup).
        13: Assign patient $j$ to mingroup.
        14: Classify using DT associated with mingroup.
    15: End For.
16: End For
End
```

More specifically, I applied my nearest consensus clustering method by running the K-means algorithm on the training data to produce 10 repeated resampled datasets to produce an agreement matrix using the input clusters. I did so in order to capture sampling bias. Each input cluster displayed the cluster output for each repeated resample. The agreement matrix, shown in fig 4.3, indicates the number of different clustering algorithms that grouped pairs of patients together. Hierarchical clustering was then applied to the agreement matrix to create the consensus clusters. Hierarchical clustering is another unsupervised approach for undertaking exploratory data analysis. It works by constructing a binary merge tree, starting with the data stored at the leaves and merging the closest subsets until the whole dataset is reached. This type of hierarchical clustering is called agglomerative hierarchical clustering (Frank Nielsen, 2016). I applied this algorithm to the agreement matrix in order to produce the consensus groups. As a result, the values of the cells in the agreement matrix were used to implement this technique utilising a further linkage measure. I allocated each patient to a distinct cluster. After identifying the largest values in the agreement matrix, I merged the points with the largest values, and the agreement matrix updated accordingly. I repeated these steps until I discovered the consensus groups. Cross validation (see Chapter 3) is an evaluation technique used to examine a model's predictive capabilities by using unseen cases to assess its accuracy. This method is implemented by separating the original data into a training set for learning the model and a test set for evaluating it, then repeatedly crossing-over the training and validation sets so that each data point is used for validation. As stated previously, I assigned test data (new case) to the closest cluster (group/consensus clusters group) using a single linkage metric and a Euclidean distance metric. Using this test data and all other data in the consensus groups, the Euclidean distance metric was computed. The new test data was assigned to the group whose point was closest to the test point. Each consensus group built its own DT model, which was then evaluated using the test data supplied to each group.

$$
\begin{array}{cc}
 & \text{To gene} \\
\text{From gene} &
\begin{bmatrix}
0 & A_{12} & A_{13} & A_{14} & \cdots & \cdots & \cdots & \cdots & \cdots & A_{1n} \\
\vdots & 0 & A_{23} & A_{24} & \cdots & \cdots & \cdots & \cdots & \cdots & A_{2n} \\
\vdots & \vdots & 0 & A_{34} & \cdots & \cdots & \cdots & \cdots & \cdots & A_{3n} \\
\vdots & \vdots & \vdots & \ddots & \ddots & & & & & \vdots \\
\vdots & \vdots & \vdots & & \ddots & \ddots & & & & \vdots \\
\vdots & \vdots & \vdots & & & \ddots & A_{ij} & & & \vdots \\
\vdots & \vdots & \vdots & & & & \ddots & \ddots & & \vdots \\
\vdots & \vdots & \vdots & & & & & \ddots & \ddots & \vdots \\
\vdots & \vdots & \vdots & & & & & & \ddots & A_{(n-1)n} \\
0 & 0 & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & 0
\end{bmatrix}
\end{array}
$$

Figure 4.3: *Agreement Matrix for Robust and Consensus Clustering (Swift et al., 2004)*

The following workflow, shown in Fig 4.4, depicts the proposed method used in this study.

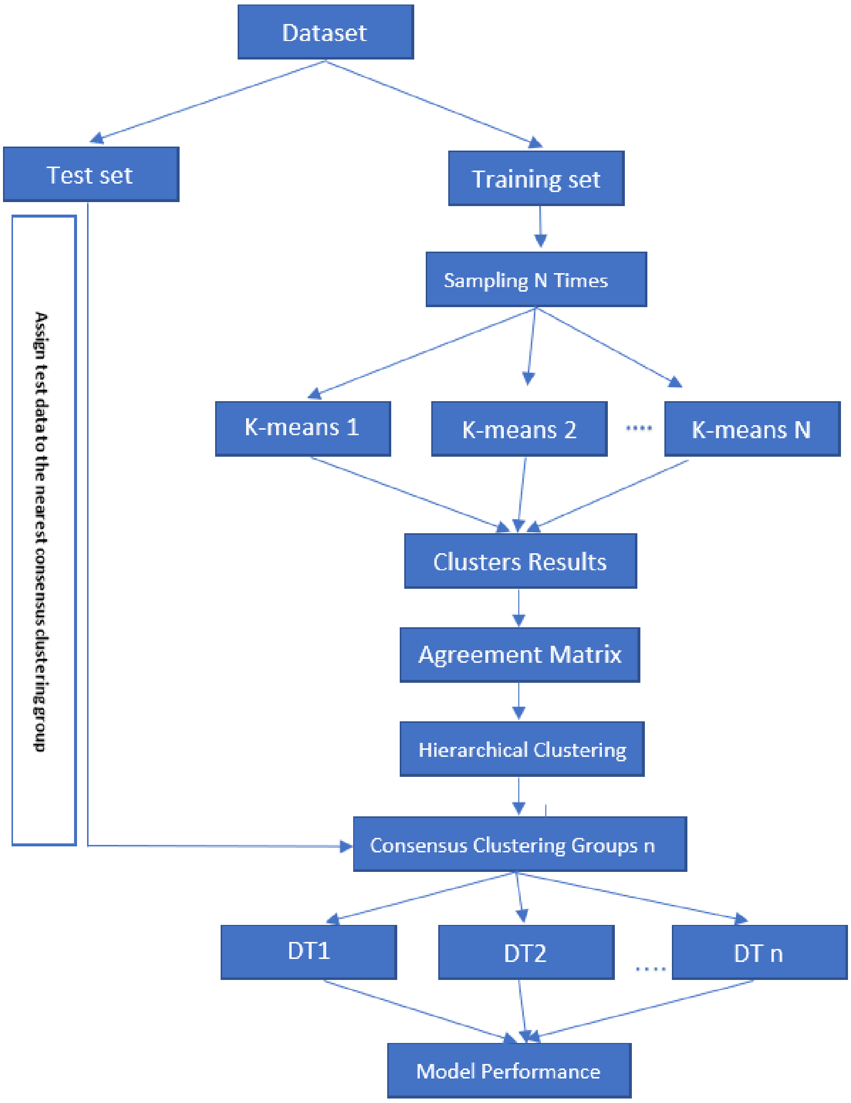


Figure 4.4:Nearest Consensus Clustering Classification Method

4.5 Datasets

I explored two datasets in this chapter: one on SSc patients and one on breast cancer patients.

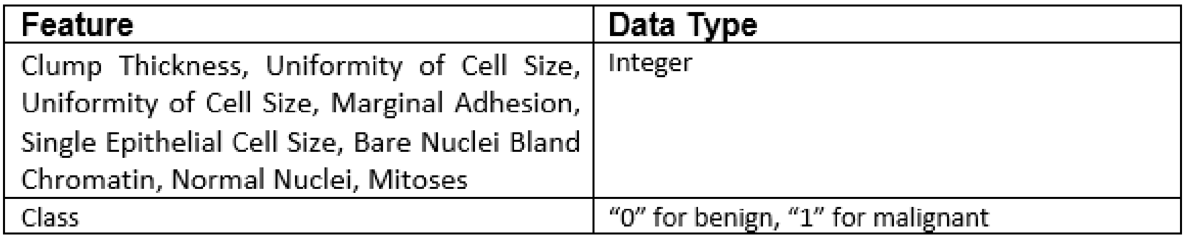
SSc Dataset

I used this dataset, provided by the Royal London Hospital (see chapter 3), to implement my new consensus clustering classification method. This dataset contained information on 677 patients.

Breast Cancer Dataset

The UCI ML repository has made this breast cancer dataset openly available. I aim to predict whether a tumour is benign or malignant using a set of 10 features and 699 patients in addition to the class. This is a multivariate dataset that was created for classification purposes. The names of the features and their data types are listed in the table below. The experiments performed on the aforementioned datasets using the proposed method are described in the following section.

Table 4.1: *Breast Cancer Features and Data Types*



Feature	Data Type
Clump Thickness, Uniformity of Cell Size, Uniformity of Cell Size, Marginal Adhesion, Single Epithelial Cell Size, Bare Nuclei Bland Chromatin, Normal Nuclei, Mitoses	Integer
Class	“0” for benign, “1” for malignant

4.6 Experiments

This section describes the experiments conducted in order to implement my proposed model using the SSc and breast cancer datasets.

SSc Dataset

As previously stated, every organ can be clinically affected by SSc; therefore, in this study, I are particularly interested in predicting whether different organ complications will occur before or after a specific threshold in order to intervene more effectively. For example, pulmonary arterial hypertension (PAH) is a significant complication of SSc; it can afflict both categories in similar proportions, and it typically arises late in the course of the disease. PAH is a progressive ailment characterised by increased blood pressure in the arteries of the lungs. It is defined by right heart catheterisation as a mean pulmonary arterial pressure not less than 25 mmHg with a pulmonary capillary wedge pressure not greater than 15 mmHg. Though the natural course of SSc-associated PAH varies across patients, in many cases it progresses to right heart failure and death. It usually presents with non-specific symptoms of exertional dyspnoea, fatigue, angina, and exertional near-syncope. SSc shows heterogeneous clinical manifestations with a wide variability in presentation, severity, and outcome: some patients deteriorate rapidly and fatally, while others experience benign symptoms. There are three key factors affecting illness susceptibility: age, gender, and ethnicity (Bosoni et al., 2016). As such, I aim to predict time to death (T2RIP) and time to PAH, a common comorbidity in patients with SSc. The aim of my proposed algorithm is to cluster the patients into consensus groups and to predict time to PAH and T2RIP for each group. The patients were selected using the following methods:

- I selected all patients from the original dataset who died within the first 5 years and all patients who were still alive or died over 5 years. The predicted class had two values: '1', representing patients who could die before 5 years, and '2', representing patients who could die after 5 years. The novel algorithm was applied to the resulting dataset in order to predict T2RIP.
- I selected all patients from the original dataset who developed PAH within the first 5 years and all patients who developed PAH after 5 years or still have not developed PAH. The predicted class had two values: '1', meaning that the patient could develop PAH within 5 years, and '2', meaning that the patient could develop PAH after 5 years. Additionally, the novel algorithm was applied on this resulting dataset in order to predict time to develop PAH.

Breast Cancer Dataset

I applied my proposed model to this dataset to classify breast cancer patients as having benign or malignant tumours. I implemented my model using the following experiments.

In addition to describing the proposed method, the goal of this chapter is to compare my novel approach, nearest consensus clustering classification, to results found through standard K-means clustering of patients, the C4.5 DT (with no clustering of patients), and the nearest K-means method (without consensus clustering). Thus, through the experiments described in this chapter, I investigated three methods:

1. Using simple K-means alone to identify clusters (with no resampling or consensus clustering) to build each DT; I call this 'nearest K-means'
2. Using a standard DT with no clustering
3. Using the novel nearest consensus clustering classification algorithm

I explored these methods based on the resulting DTs, cluster membership, predictive accuracy, and Kaplan Meier curves for:

a. the SSc data for predicting time to PAH.
b. the SSc data for predicting T2RIP.
c. the breast cancer data for predicting tumour type.

I then performed the following analyses:

d. I performed full sensitivity analyses of these methods.
e. I performed a small follow-up data analysis on the discovered groups within the clinical context.
f. I explored the effect of changing the number of clusters (K) on accuracy.
g. I compared my proposed approach with other, similar combinations of clustering and classifiers.

Finally, I briefly compared my approach with well-known methods including hierarchical clustering, partition around medoids (PAM), and support vector machines (SVM) methods. MATLAB software was implemented each of these techniques. I compared the methods described below to my method.

1. **Hierarchical clustering using decision tree method:**

Hierarchical clustering partitions a dataset into groups using a dendrogram tree structure, as described above. This was combined with a DT algorithm to predict T2RIP, to predict time to develop PAH, and to classify breast cancer patients.

2. **PAM using decision tree method:**

PAM clustering (k-medoid algorithm) is similar to the K-means method in that it splits the dataset into *K* groups; however, in PAM clustering, medoids (rather than centroids) are represented by data points. These data points correspond to the most centrally located point in each cluster (Al Abid & Mottalib,2012). This algorithm was combined with a DT algorithm to predict T2RIP, to predict time to develop PAH, and to classify breast cancer patients.

3. Standard SVM method:

The supervised ML model (SVM) works by transforming data and conducting simple scaling so that classes are linearly separable. SVM is often considered the most consistently accurate classifier. The disadvantage of this algorithm is the complexity involved in determining the number of support vectors (Cristianini & Taylor, 2000). This algorithm was implemented to predict T2RIP, to predict time to develop PAH, and to classify breast cancer patients.

4.7 Results

In section, I examine the outcomes of my experiments after having described the proposed method and the experiments I conducted for this study. Boxplots, a simple graphical technique used to describe results, are employed to interpret and compare my results. The minimum value, maximum value, and median of a set of data, such as error rate across my experiments, are represented by boxplots. Additionally, t-tests were used to compare the significance difference between two means.

A) Systemic sclerosis: Time to develop pulmonary arterial hypertension

I ran the C4.5 DT (without clustering), nearest K-means (without consensus), and nearest consensus cluster classification algorithms on the SSc data in order to predict time to develop PAH. The following plot (Figure 4.5) shows the results of these experiments as well as the results of each individual cluster model on all of the test data (K1, K2, and K3). It is notable that each individual clustering model classified the test data worse than ones that attempted to model all clusters. Additionally, the standard DT and nearest K-means methods produced a better and less variable set of errors. Nearest consensus cluster classification performed better than all other algorithms, with lower error rates and reduced variance. This method performed significantly better than the nearest K-means method ($p = 0.040 < 0.05$), indicating that sampling bias should be addressed when identifying patient subgroups.

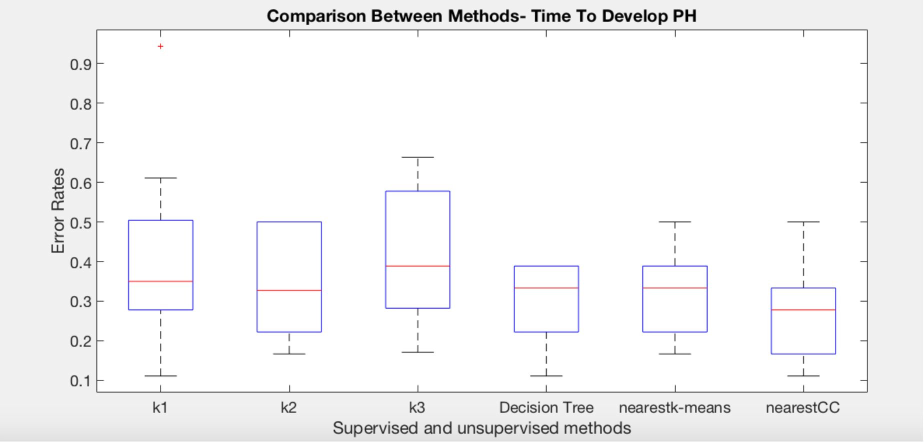
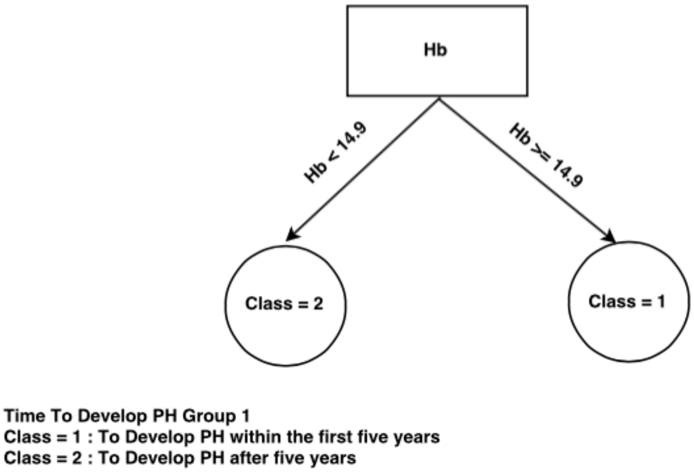


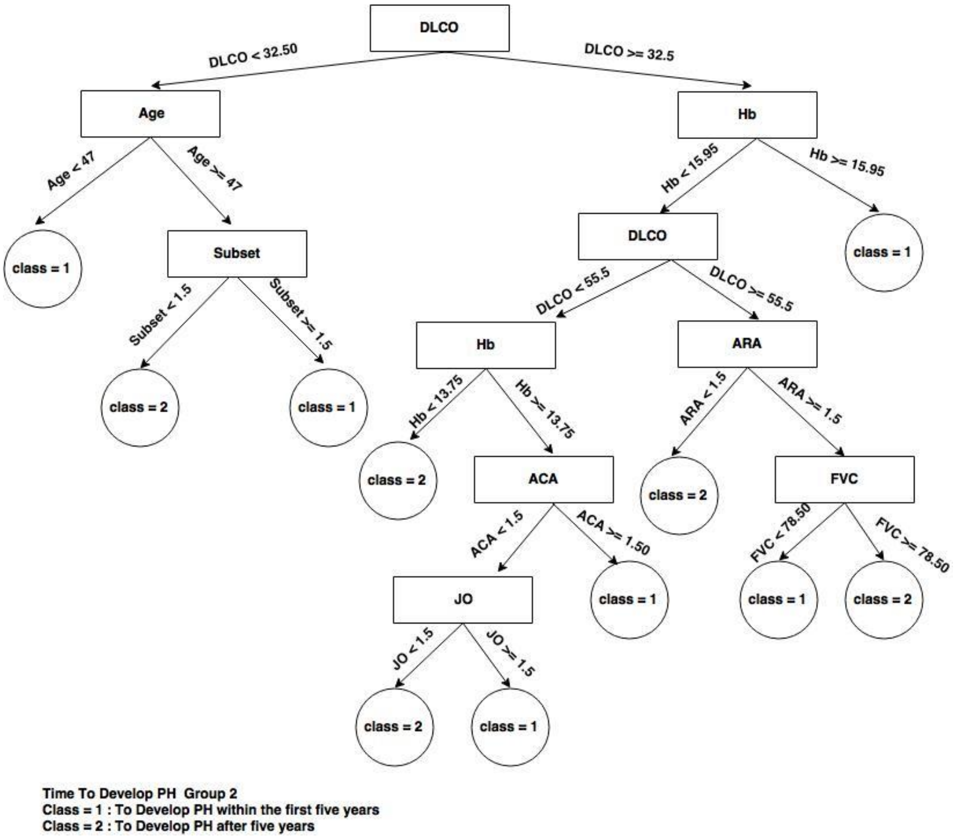
Figure 4.5: *Comparison of K-means, Decision Tree, Nearest K-means, and Nearest CC for Time to Develop Pulmonary Arterial Hypertension Class in Systemic Sclerosis Dataset*

The DTs inferred from each consensus cluster found in the SSc dataset when time to develop PAH class needs to be predicted (Figures 4.6–4.8) were very different, indicating that there was a different set of required criteria for each subset of patients that was discovered. For example, the group 1 DT was considerably smaller than the group 2 or group 3 DTs, and all the DTs involved different combinations of important variables. This highlights the necessity of separating out these cohorts of patients when diagnosing. For instance, for group 3, knowing the DLCO, age, and FVC test result had more of an effect on predicting time to develop PAH, whereas in group 1, knowing only the haemoglobin (Hb), ACA, and others had more of an effect on predicting time to develop PAH. Fig 4.6 displays a very simple DT that only relies on the Hb variable; therefore, for group 1, time to develop PAH can be predicted based on Hb attribute values.



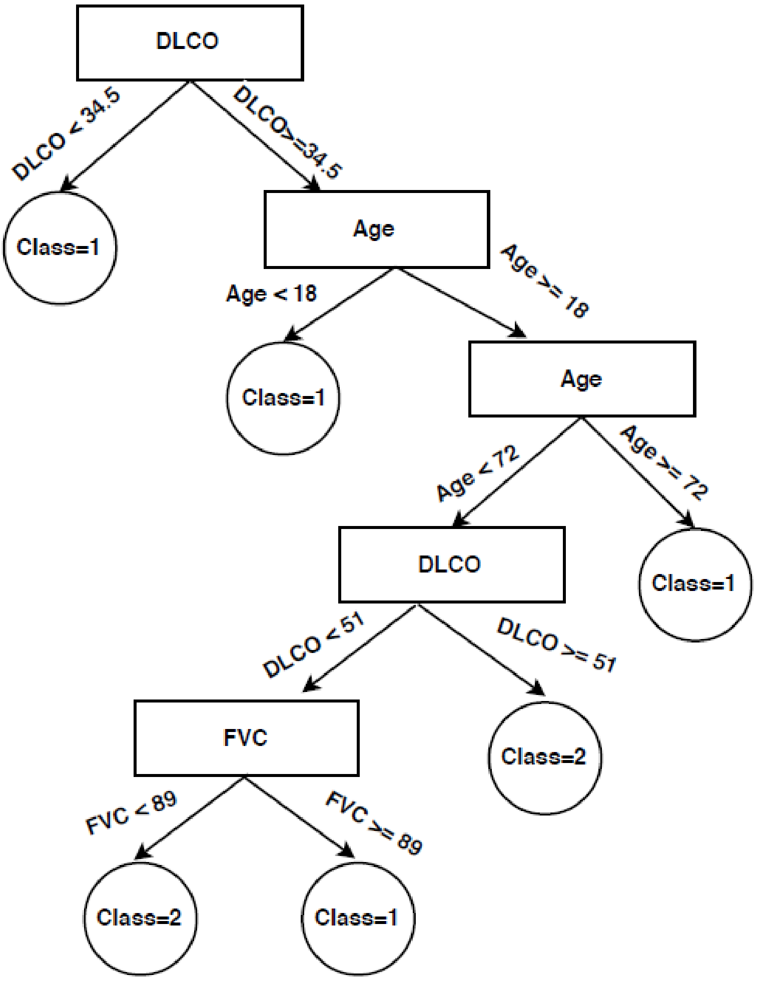
Time To Develop PH Group 1
Class = 1 : To Develop PH within the first five years
Class = 2 : To Develop PH after five years

Figure 4.6: *Consensus Clustering Decision Tree for Group 1 in SS Dataset and Time to Develop Pulmonary Arterial Hypertension Class.*



Time To Develop PH Group 2
Class = 1 : To Develop PH within the first five years
Class = 2 : To Develop PH after five years

Figure 4.7: *Consensus Clustering Decision Tree for Group 2 in SS Dataset and Time to Develop Pulmonary Arterial Hypertension Class.*

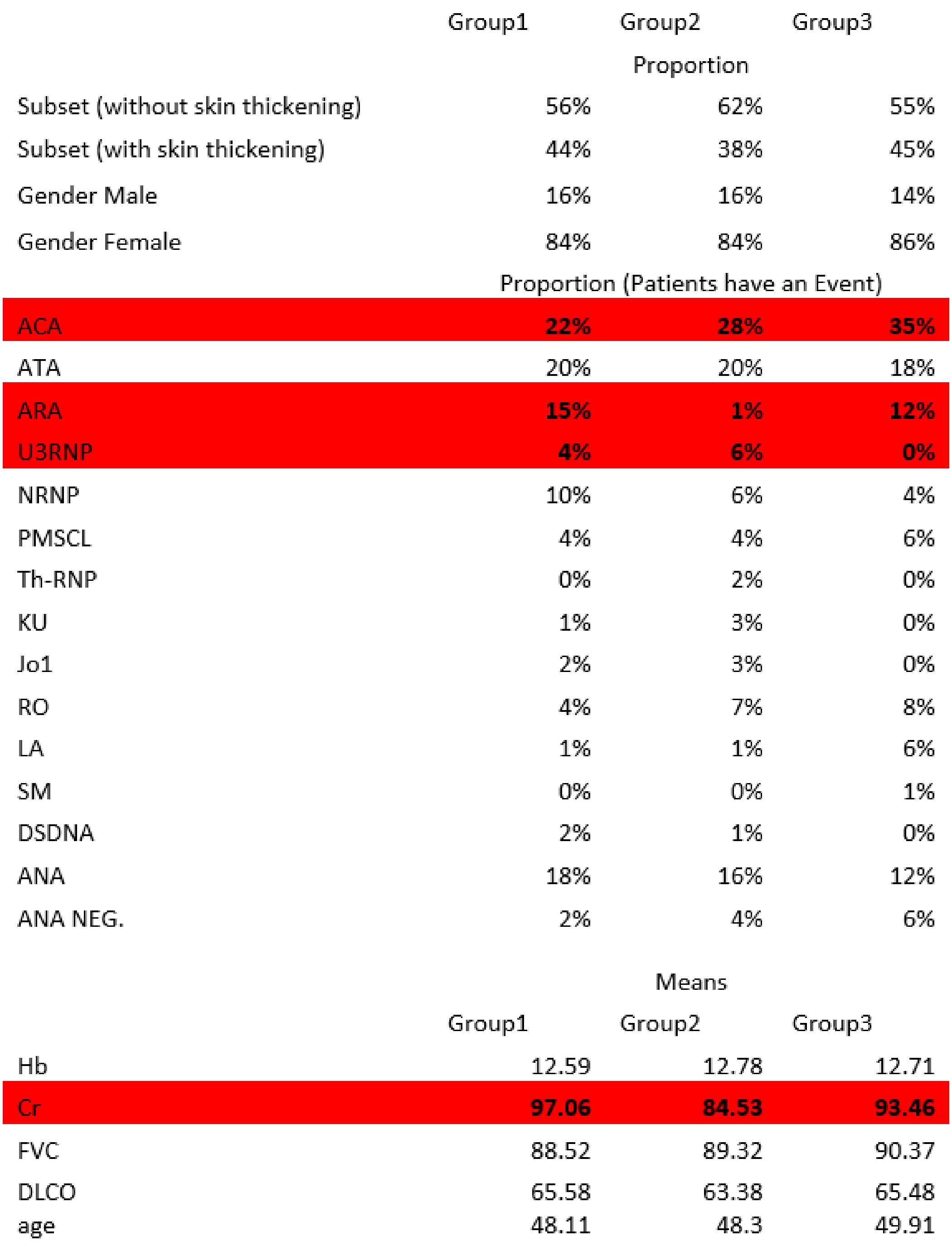


Time To Develop PH For Group3
Class = 1 : To Develop PH within the first five years
Class = 2 : To Develop PH after 5 years

***Figure 4.8**: Consensus Clustering Decision Tree for Group 3 in SS Dataset and Time to Develop Pulmonary Arterial Hypertension Class.*

There were notable differences between the attributes in each discovered consensus cluster (Table 4.2). Serum creatinine (Cr) level, the value indicating the measure of creatinine in that test, was smaller for group 2 than for groups 1 or 3. Cr values greater than 90 – such values were found in groups 2 and 3 – are normal, but values less than 90 are not normal. The reference range for the time period for Cr was 60–97 μmol/L. Interestingly, these features did not appear in the DTs, perhaps because they were already separated by the identification of the different subgroups. By identifying these different subgroups and exploring their characteristics, I can better understand how they differ and what focused tests may be appropriate for different patients when determining prognoses. By identifying the characteristics of each consensus cluster, I can identify the likelihood that patients belong to any of these cohorts and apply more appropriate clinical tests, as identified using the cohort-specific DTs. This is essentially what the algorithm does when in the testing phase.

Table 4.2: *Proportion/Means Values for SS Attributes in CC (Time to Develop PAH)*



	Group1	Group2	Group3
		Proportion	
Subset (without skin thickening)	56%	62%	55%
Subset (with skin thickening)	44%	38%	45%
Gender Male	16%	16%	14%
Gender Female	84%	84%	86%
		Proportion (Patients have an Event)	
ACA	**22%**	**28%**	**35%**
ATA	20%	20%	18%
ARA	**15%**	**1%**	**12%**
U3RNP	**4%**	**6%**	**0%**
NRNP	10%	6%	4%
PMSCL	4%	4%	6%
Th-RNP	0%	2%	0%
KU	1%	3%	0%
Jo1	2%	3%	0%
RO	4%	7%	8%
LA	1%	1%	6%
SM	0%	0%	1%
DSDNA	2%	1%	0%
ANA	18%	16%	12%
ANA NEG.	2%	4%	6%
		Means	
	Group1	Group2	Group3
Hb	12.59	12.78	12.71
Cr	**97.06**	**84.53**	**93.46**
FVC	88.52	89.32	90.37
DLCO	65.58	63.38	65.48
age	48.11	48.3	49.91

I then conducted disease-free survival analysis. The Kaplan-Meier estimator, also known as the product limit estimator, is a non-parametric statistical method used to estimate the survival function in reference to an event of interest, such as death or a disease complication (Goel M.K. and Khanna P. and Kishore, 2010). The estimator is plotted over time to obtain the Kaplan-Meyer curve, which comprises a series of horizontal steps of declining magnitude that, when a large sample is taken, approaches the true survival function for the population under investigation. This curve can be estimated easily if a patient group is followed until death by computing the fraction of patients surviving at each time point. In most cases, however, a number of patients tend to drop out for various reasons. Nevertheless, Kaplan-Meyer analysis allows this information from both censored and uncensored observations to be considered. The dependent variable is composed of two parts: the time to the event and the event status, which records whether or not the event of interest has occurred. Censored data is data for which the event is only partially known because it has not yet occurred. For example, in the SSc dataset, I may only know that a patient has not developed PAH for at least X years at a given point in time. The Kaplan-Meier curve is defined as the probability of surviving for a given length of time while considering time in many small intervals, taking only three weak hypotheses into account (Altman, D.G., 1990). It must be assumed that the censored patients are characterised by the same survival prospects as those who continued to be followed, that the survival probabilities are the same for patients recruited early in the study and those recruited later, and that the event of interest happens at the specified time (Goel M.K. and Khanna P. AND Kishore, 2010).

I conducted a survival analysis to determine how long after diagnosis patients in the discovered subgroups died or developed a disease-associated internal organ complication. By grouping subjects based on nearest consensus clustering classification, I could then analyse whether the discovered clusters were able to separate SSc patients into subpopulations showing different symptoms and disease progressions in order to help physicians to make more informed diagnoses and carry out more focused interventions.

The following graph shows the percentage of patients who survived from that organ complication on the y-axis, while on the x-axis, the time to develop PAH is measured in months. The graph shows the Kaplan-Meyer curves for the three main clusters: cluster 1 is blue, cluster 2 is green, and cluster 3 is yellow. The graph clearly shows that 18% of patients in group 3 and about 10% of patients in group 1 were affected by PAH after 120 months.

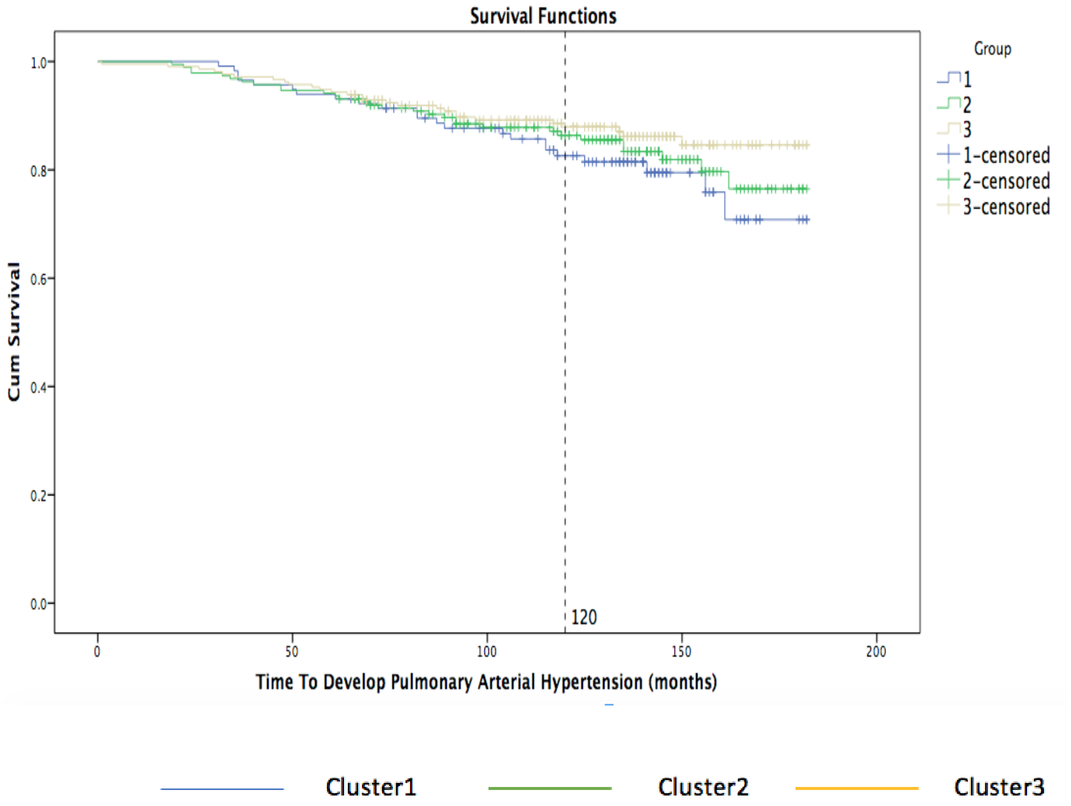


Figure 4.9: *Kaplan-Meyer Curves by Nearest Consensus Clustering on Time to Develop Pulmonary Arterial Hypertension Dataset. With time to develop pulmonary arterial hypertension in months on the x-axis and percentage of patients survived from that organ complication on the y-axis, the graph illustrates the survival curves obtained by grouping patients based on nearest consensus clustering.*

B) Systemic sclerosis: Time to death

I repeated the experiments described above to predict T2RIP. The following plot (Figure 4.10) displays the results of these experiments as well as the results of the application of each individual cluster model to all of the test data (K1, K2 and K3). Note that these groups are not the same as those for the time to PAH experiments, as different data was selected. The following boxplot (Figure 4.10) shows that nearest consensus clustering classification performed better than the nearest K-means method, although the latter has less variation ($p = 0.041 < 0.05$).

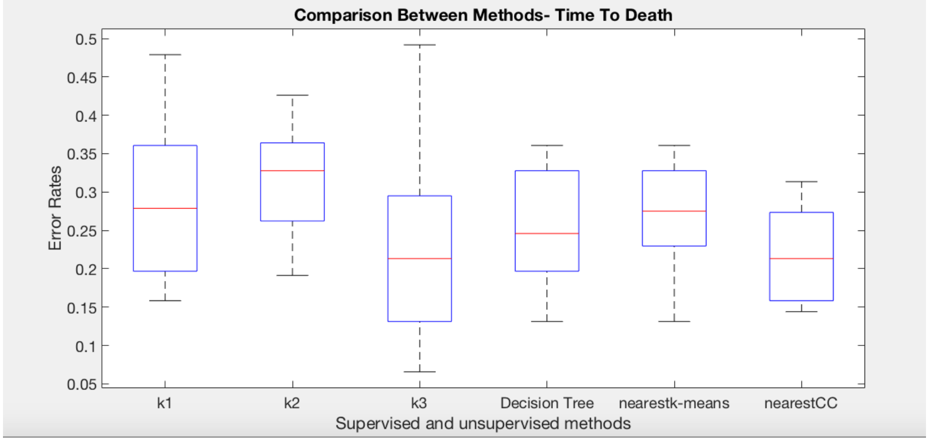


***Figure 4.10**: Comparison of K-means, Decision Tree, Nearest K-means, and Nearest CC for Time to Death Class in Systemic Sclerosis Dataset*

The DTs inferred from each consensus cluster found in the SSc dataset to predict T2RIP (Figures 4.11–4.13) were very different, indicating that different sets of required criteria have been discovered for each patient subset. These various consensus DTs could improve clinics' understanding of the disease and enhance personalised medicine.

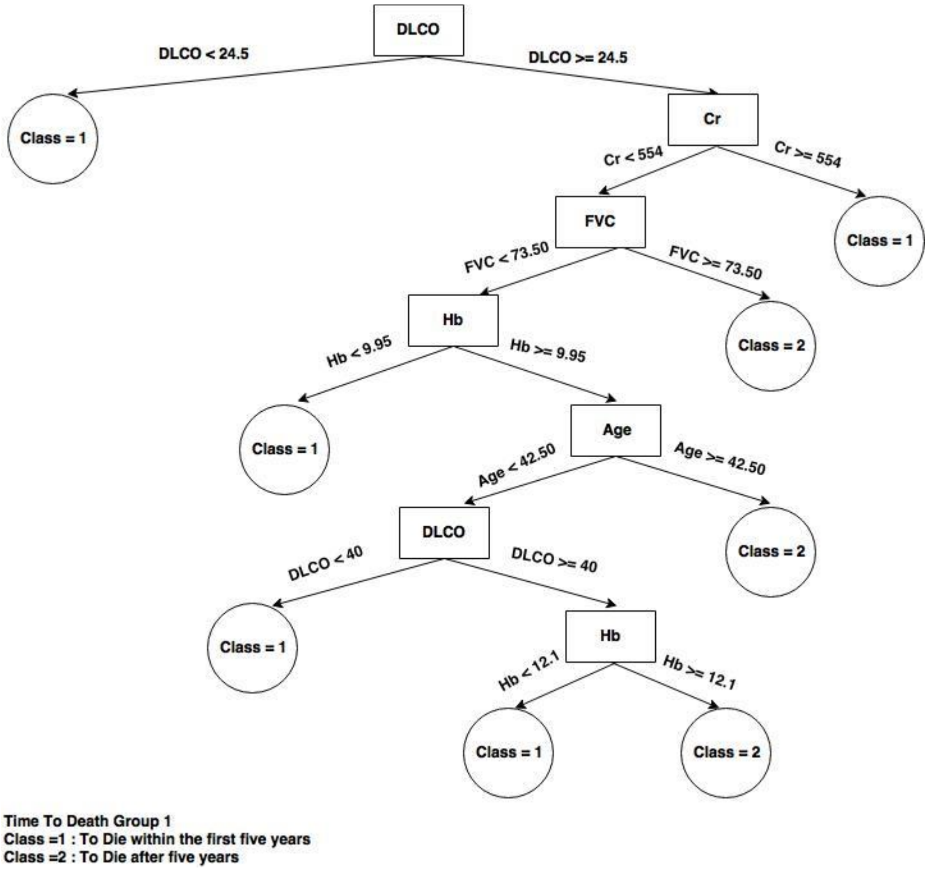


***Figure 4.11**: Consensus Clustering Decision Tree for Group 1 in SS Dataset and Time to Death class*

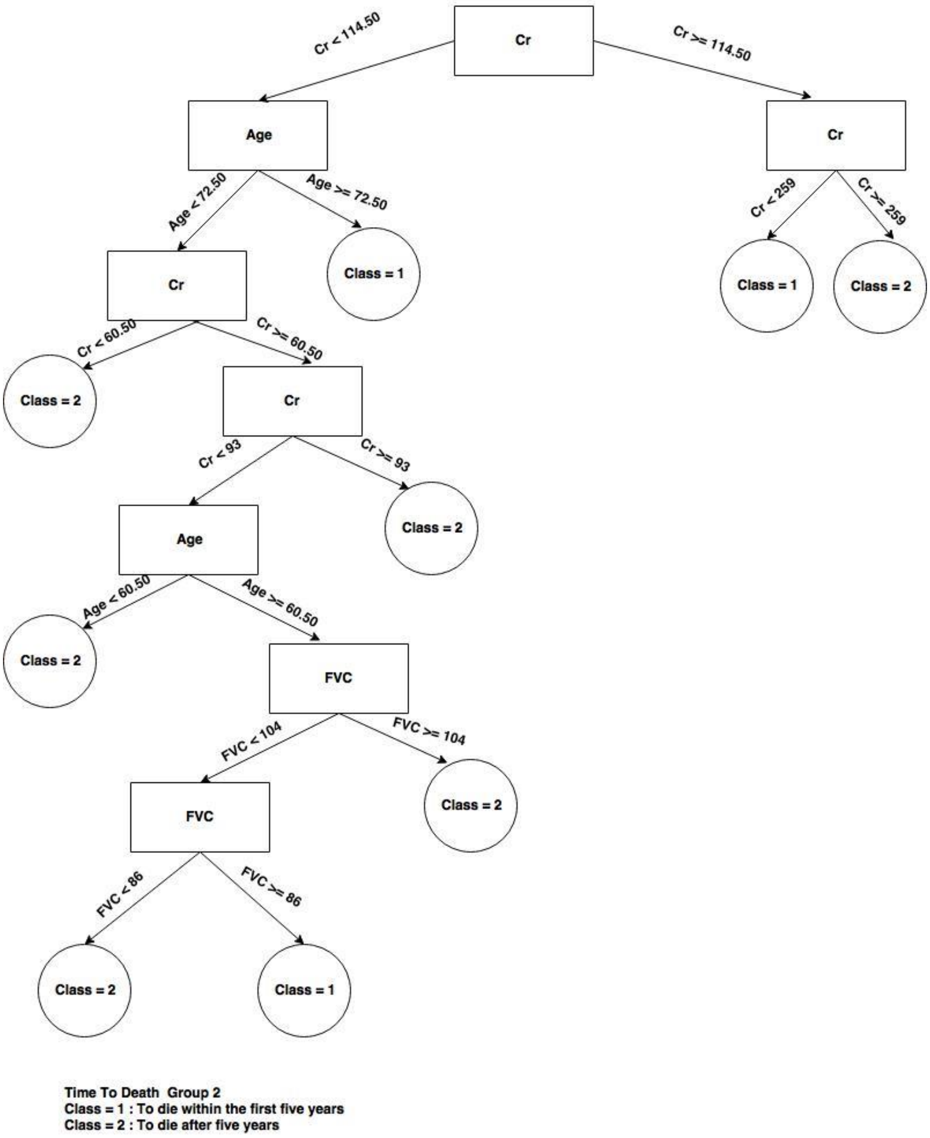


Figure 4.12*: Consensus Clustering Decision Tree for Group 2 in SS Dataset and Time to Death Class*

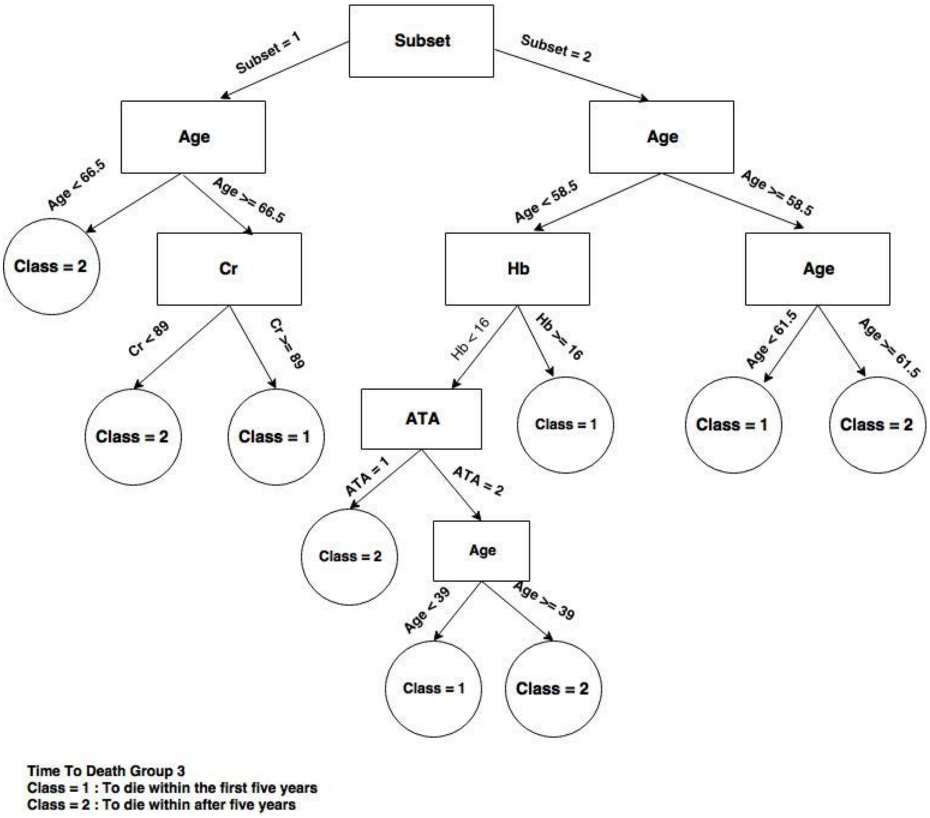


***Figure 4.13**: Consensus Clustering Decision Tree for Group 3 in SS Dataset and Time to Death Class*

Regarding survival analysis, the following graph shows that almost 35% of patients from group 1 died after 110 months, while 15% of patients from groups 2 and 3 died after 110 months.

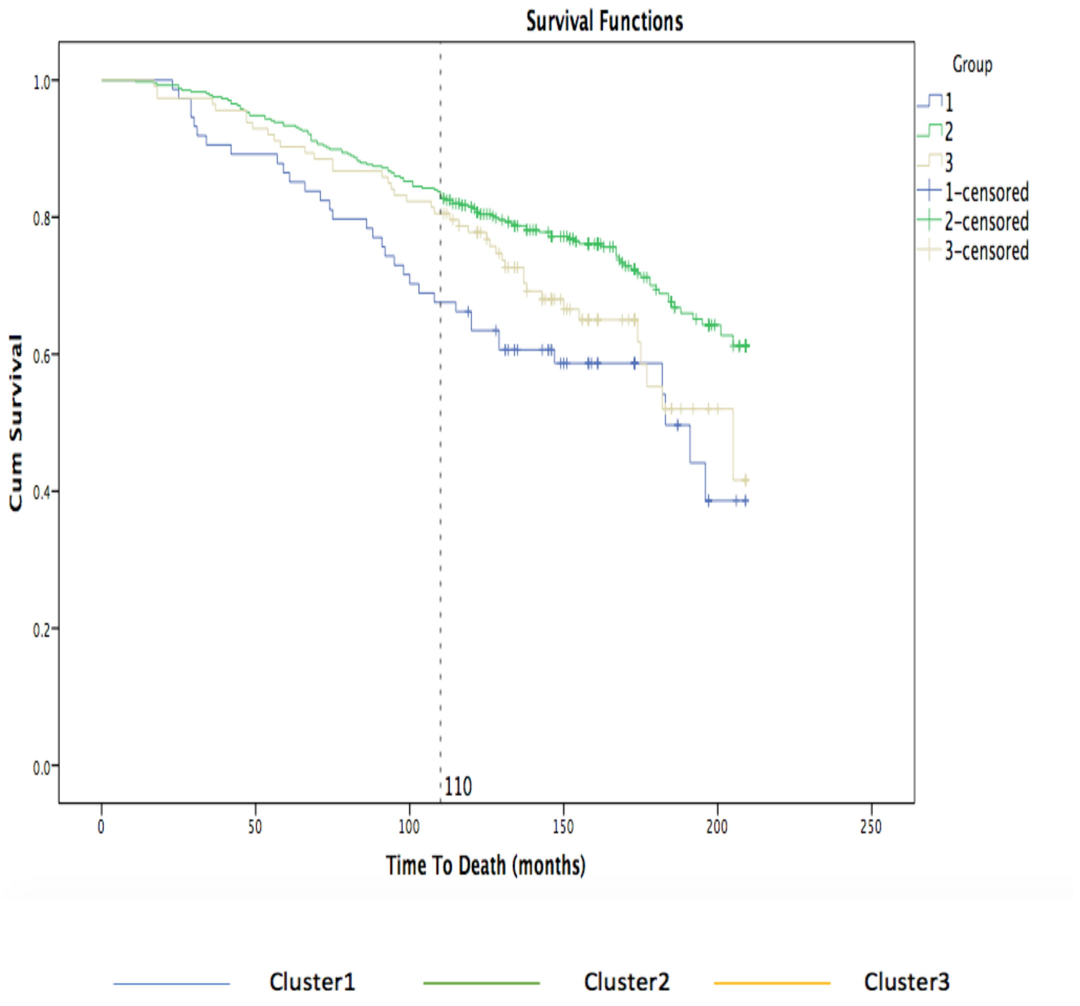
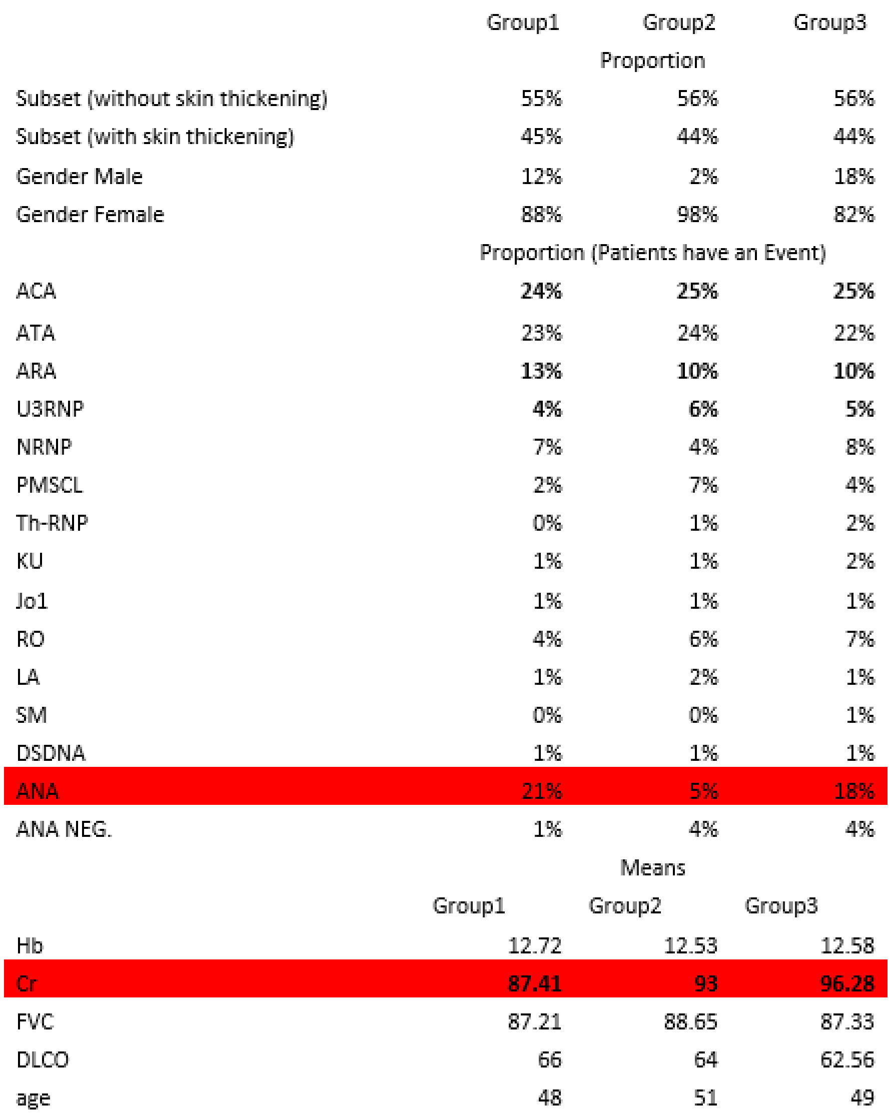


Figure 4.14*: Kaplan-Meyer Curves by Nearest Consensus Clustering on Time to Death Dataset. With time to death in months on the x-axis and percentage of surviving patients on the y-axis, the graph illustrates the survival curves obtained by grouping patients based on nearest consensus clustering.*

Again, there were notable differences between the attributes of each discovered consensus (Table 4.3). Cr, the value indicating the measure of creatinine in that test, was greater in group 1 than in groups 1 or 3. Cr values greater than 90 – such values were found in groups 2 and 3 – are normal, but values less than 90 are not normal. The baseline used to distinguish among the groups is whether Cr was normal or not. Additionally, DLCO was smallest in group 3.

Table 4.3: *Proportion/Means Values for SS Attributes in CC (Time to Death)*



	Group1	Group2	Group3
		Proportion	
Subset (without skin thickening)	55%	56%	56%
Subset (with skin thickening)	45%	44%	44%
Gender Male	12%	2%	18%
Gender Female	88%	98%	82%
		Proportion (Patients have an Event)	
ACA	**24%**	**25%**	**25%**
ATA	23%	24%	22%
ARA	**13%**	**10%**	**10%**
U3RNP	**4%**	**6%**	**5%**
NRNP	7%	4%	8%
PMSCL	2%	7%	4%
Th-RNP	0%	1%	2%
KU	1%	1%	2%
Jo1	1%	1%	1%
RO	4%	6%	7%
LA	1%	2%	1%
SM	0%	0%	1%
DSDNA	1%	1%	1%
ANA	21%	5%	18%
ANA NEG.	1%	4%	4%
		Means	
	Group1	Group2	Group3
Hb	12.72	12.53	12.58
Cr	**87.41**	**93**	**96.28**
FVC	87.21	88.65	87.33
DLCO	66	64	62.56
age	48	51	49

C) Breast cancer

I also applied my method to the freely available breast cancer dataset from the UCI repository. K-means, DT, nearest K-means and nearest CC classifications were applied in order to predict whether tumours were malignant or benign. Again, the results, as shown in Fig. A4, indicated that nearest CC classification performed better than the K-means or standard DT methods.

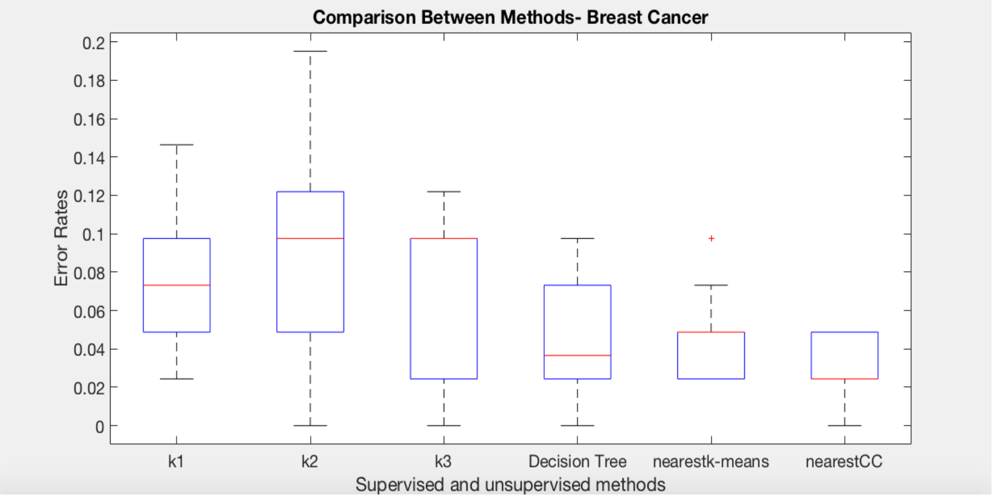


***Figure 4.15**: Comparison of K-means, Decision Tree, Nearest K-means, and Nearest CC Classification for Breast Cancer Dataset*

Fig 4.16 shows the DTs that predicted breast cancer in group 1. Breast cancer, whether benign or malignant, could easily predicted for all patients within group 1 through analysis of cell shape and cell size. Additionally, fig 4.17 shows that, for group 2 patients, thickness must be known in order to predict breast cancer.

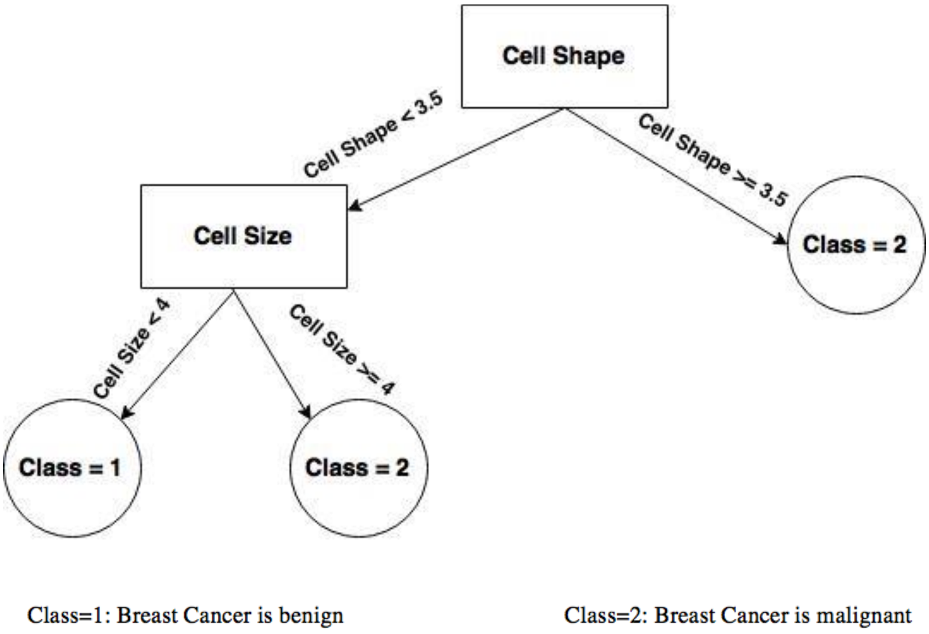


***Figure 4.16**: Consensus Clustering DT for Breast Cancer Prediction Group 1*

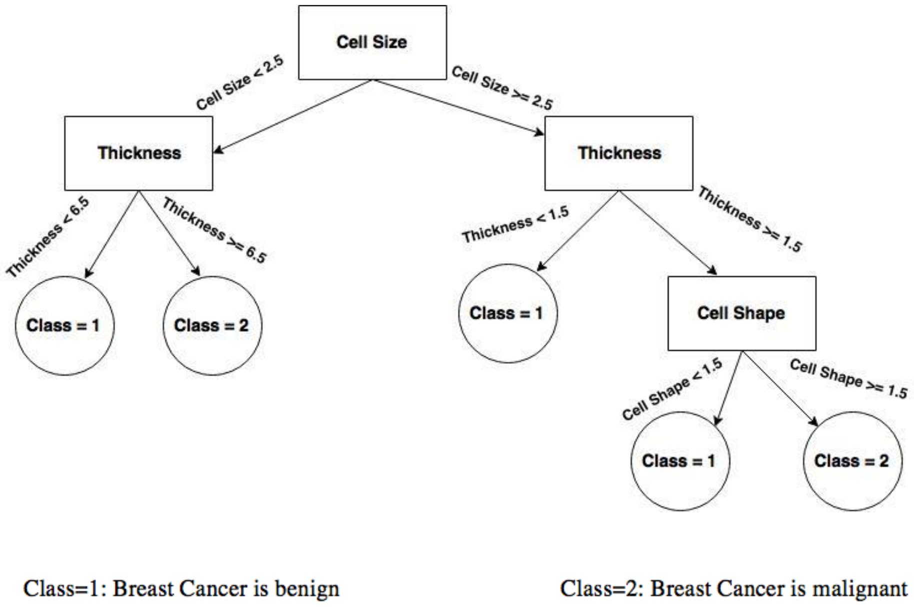


Figure 4.17: *Consensus Clustering DT for Breast Cancer Prediction Group 2*

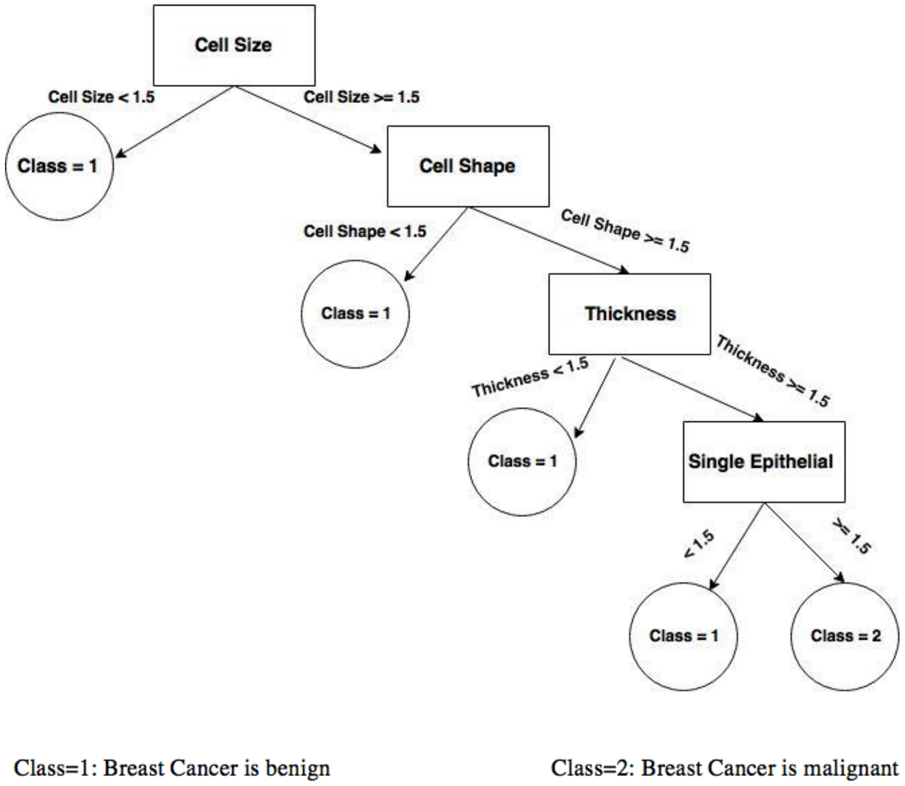


Figure 4.18*: Consensus Clustering DT for Breast Cancer Prediction Group 3*

Table 4.4 shows the variation of attributes across each group. Thickness values in group 1 were greater than those in groups 2 or 3. Additionally, cell size values for group 3 were smaller than those in groups 1 or 2.

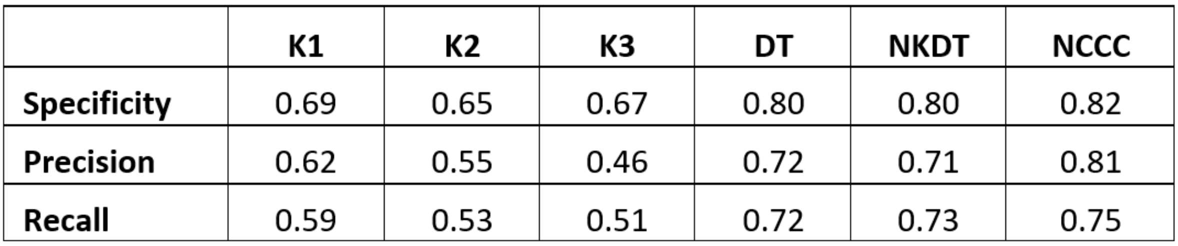
***Table 4.4**: Means for Breast Cancer Attributes in Consensus Clustering*

Attributes	Group 1	Group 2	Group 3
Thickness	5.10	4.58	4.14
CellSize	3.87	3.14	2.97
CellShape	4.09	3.22	2.76
Marginal	3.27	2.72	2.65
Epithelial	3.74	3.60	2.86
BChromation	4.04	3.77	2.99
NormalNucleoli	3.77	3.14	2.42
Mitoses	1.97	1.85	1.33

- **Sensitivity Analysis**

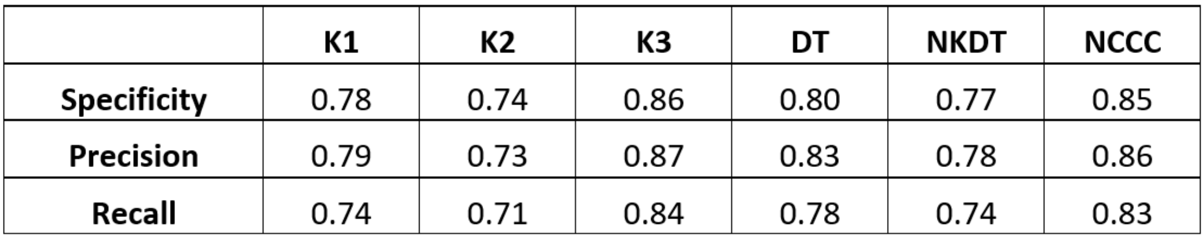
Specificity, precision, and recall metrics were used to evaluate the results. I computed all of these measures for K-means, DT, nearest K-means, and nearest CC classifications for time to develop PAH and breast cancer dataset results. The following tables display the results. It is notable that NCCC performed well compared to other methods.

***Table 4.5**: Metrics for three K-means Groups – Decision Tree, Nearest K-means, and Nearest CC –for Time to Develop Pulmonary Arterial Hypertension Class*



	K1	K2	K3	DT	NKDT	NCCC
Specificity	0.69	0.65	0.67	0.80	0.80	0.82
Precision	0.62	0.55	0.46	0.72	0.71	0.81
Recall	0.59	0.53	0.51	0.72	0.73	0.75

***Table 4.6**: Metrics for Three K-means Groups – Decision Tree, Nearest K-means, and Nearest CC – for Breast Cancer Dataset*



	K1	K2	K3	DT	NKDT	NCCC
Specificity	0.78	0.74	0.86	0.80	0.77	0.85
Precision	0.79	0.73	0.87	0.83	0.78	0.86
Recall	0.74	0.71	0.84	0.78	0.74	0.83

- **Impact of Different Number of Clusters (K)**

I briefly explored the effect of different values of K (when using K-means clustering) on accuracy. I ran nearest consensus cluster classification on the SSc data in order to predict time to develop PAH and T2RIP five times for each class. The following two plots show the results of these experiments and those of each individual consensus cluster classification model on all of the test data (K = 3, K = 4, K = 5, K = 7, K = 10). Regarding time to develop PAH, it is notable that nearest consensus cluster classification when K was equal to 3 (NCC3) and when K was equal to 4 (NCC4) classified the test data quite similarly to the others, and NCC3 and NCC4 performed better than NCC5 or NCC7. NCC10 improved error and grown variation, but it had noise. Regarding T2RIP, nearest consensus cluster classification when K was equal to 4 (NCC4) performed better and had less variation than when K was equal to 3 (NCC3). Additionally, NCC4 classified the test data better than NCC5, NCC7, or NCC10.

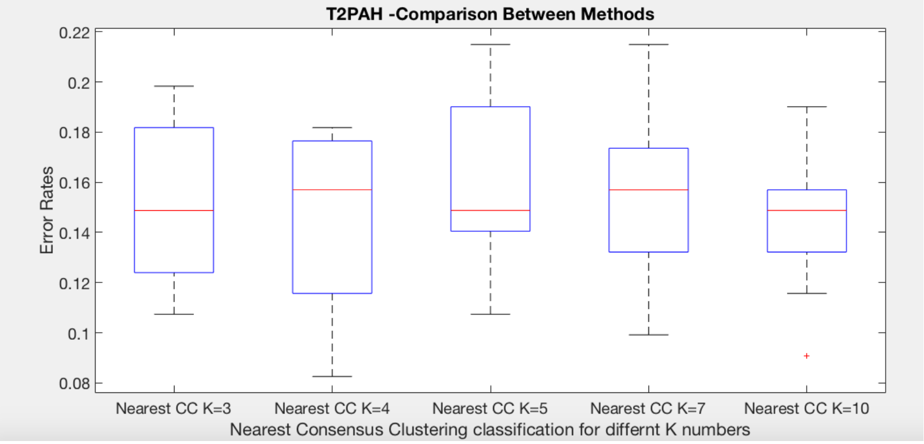


Figure 4.19: *Comparison of Nearest CC Classification for Time to Develop Pulmonary Arterial Hypertension Class With Different Values of K*

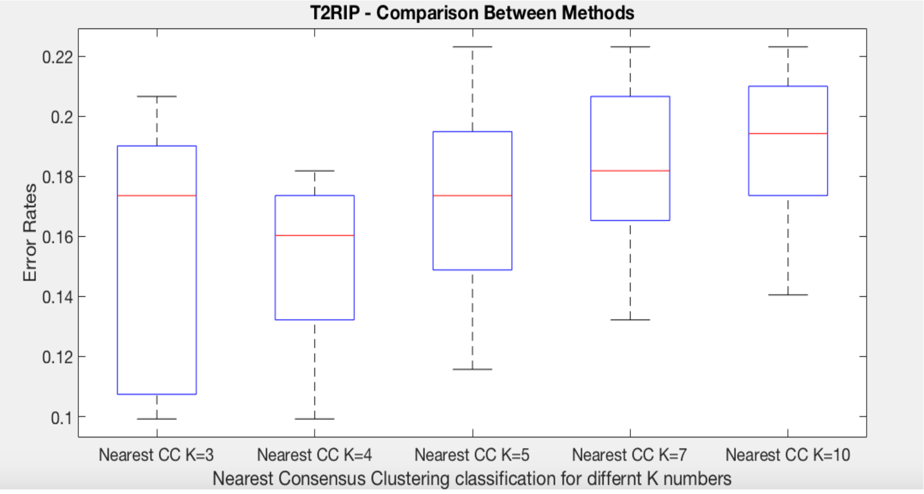
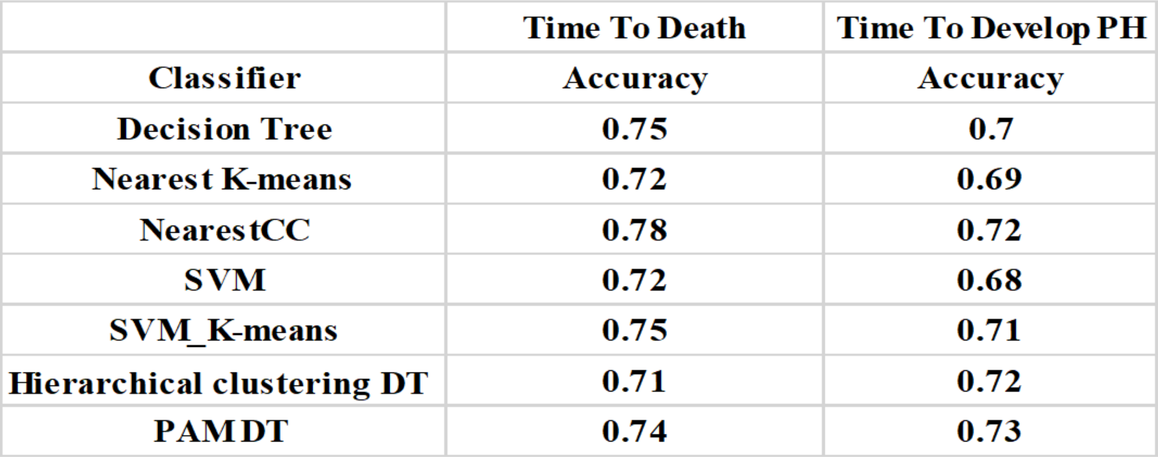


***Figure 4.20**: Comparison of Nearest CC Classification for Time to Death Class with Different Values of K*

- **Comparison to Other Clustering/Classifiers**

Finally, I briefly compared my new method with some other cluster–classifier combinations, including SVM run individually, SVM merged with K-means, hierarchical clustering DT, and PAM DT, in order to confirm whether or not the proposed method performed better. The following table displays the results. I repeated the experiment to test all of these classifiers using the model.

***Table 4.7:** Accuracy Comparison Between the Proposed Algorithm and Others*



	Time To Death	Time To Develop PH
Classifier	Accuracy	Accuracy
Decision Tree	0.75	0.7
Nearest K-means	0.72	0.69
NearestCC	0.78	0.72
SVM	0.72	0.68
SVM_K-means	0.75	0.71
Hierarchical clustering DT	0.71	0.72
PAM DT	0.74	0.73

4.8 Conclusions

In this chapter, I have tested a set of algorithms on the SSc and breast cancer datasets both to identify subgroups of patients and to diagnose them based on these subgroups. The results illustrate the issues associated with ignoring the existence of patient subgroups (doing so leads to higher error rates) and with using standard clustering methods such as k-means (doing so results in higher variance in errors due to sample variance and method bias). This chapter introduces a novel approach that exploits consensus clustering methods and single-linkage distance metrics to address these issues. My method, nearest consensus clustering classification, integrates DTs, consensus clustering, and single-linkage metrics. It improved classification and reduced variance when tested on breast cancer data from the UCI repository and an SSc dataset from the Royal Free Hospital in London. Clinics could use this new model to cluster patients and discover key features of each group to classify patients more confidently. However, my novel approach only addresses the prediction of a single disease at a given time, whereas in the real world patients may be suffering from multiple comorbidities simultaneously. Thus, it is essential to address concerns around the use of several labels (comorbidities). It is also essential to discuss another method for clustering patients into groups, such as latent class analysis (LCA), which groups patients by identifying latent factors. I therefore needed to devise a novel approach that combines latent class analysis with a multi-label classification (MLC) model to predict multiple complications by discovering hidden factors. In the next chapter, I describe MLC and latent class models that cluster patients within groups. A new algorithm that uses MLC and a latent class model is described in the next chapter.

Chapter 5

Latent Class Multi-Label Classification to Identify Subclasses of Disease for improved Prediction

5.1 Introduction

In medicine, patients might suffer from multiple disease complications. Disease subtyping can help to develop personalised treatments by grouping patients into subgroups based on their data. Thus, models that are tailored to individuals could improve both prediction of multiple complications and understanding of underlying disease characteristics. Therefore, this chapter first investigates the significance of latent class models as well as MLC problems, in which a patient may belong to numerous classes simultaneously. Additionally, it describes the phenomenon of the latent class model, and it explains the MLC problem and current methods for resolving it. Second, this chapter discusses my proposed method for addressing the MLC problem as well as the datasets used and experiments conducted. This chapter concludes by presenting and interpreting the results of applying this method to my datasets. This chapter was published in *Proceedings of the 2019 IEEE 32nd International Symposium on Computer-Based Medical Systems* (CBMS).

5.2 Overview

Healthcare organisations must find better methods to assist diagnosis that are both accurate and explainable. Machine learning classifiers that can exploit huge amounts of historical patient data are a promising technology with which to achieve this. Their aim is to accurately predict a class label for new patients (e.g. a diagnosis or risk factor) based on historical data. However, in some situations, patients might belong to more than one class. For example, a patient might have both diabetes and cancer (Nareshpalsingh & Modi, 2017). In this case, MLC, wherein multiple class labels can be assigned to a single patient’s data, can be employed. MLC is a challenging task in ML as it requires the prediction of more than one class. Previous research has demonstrated MLC’s effectiveness and robustness. Dharmadhikari et al. (2012) observed that better accuracy is obtained when MLC algorithm adaption is used (see Section 4). Additionally, Prajapati et al. introduced MLC and associated evolution metrics, suggesting that algorithm adaptation is the best option for MLC (Prajapati et al., 2012). Dhongade et al. reviewed MLC and observed that the approach is mostly used for text categorisation and medical diagnosis. This paper introduced ML-KNN and showed that it is better than other established algorithms (Dhongade et al., 2014). MLC aims to predict all classes to which a patient belongs, and it may also help to identify the relationships between classes. A novel model for MLC based on Bayesian networks was introduced in order to predict all classes whilst simultaneously finding correlations among them. This model performed better than binary classification methods (Alessandro et al., 2013). Damien Zufferey et al. (2015) employed multilabel algorithms on chronic disease data to simultaneously predict different chronic illnesses. This model assists clinicians in diagnosing, understanding, and treating disorders (Zufferey et al., 2015). Giorgio Corani and Mauro Scanagatta (2016) tried to predict air pollution using MLC methods. They designed a MLC model dependent on a Bayesian network. The proposed model performed well, and it allowed experts to make better decisions regarding air pollution (Giorgio & Mauro, 2016). In traditional Chinese medicine, diagnosis aided by ML algorithms enables practitioners to utilise sophisticated medical knowledge and make decisions more efficiently. Zhou et al. (2018) proposed a model for diagnosing diseases in traditional Chinese medicine based on a MLC method. The results demonstrated the model's potential to enhance decision-making within the clinical diagnostic system (Zhou et al., 2018). Although many researchers have worked to enhance NB performance in classification problems, less research has been done on multilabel NB classification. Shouman et al. explored the effectiveness of K-means as a clustering method for improving supervised learning techniques like NB. Results showed that integrating a clustering method with NB could enhance accuracy (Shouman et al., 2012). The NB method is considered important for medical diagnosis as it can provide accurate results and reveal hidden information between the variables (Vembandasamy et al., 2015). Kabir et al. claimed that higher accuracy can be produced when datasets are split into subgroups, wherein each group has similar intra-group characteristics. They focused on improving the classification accuracy of NB by clustering the dataset using the K-means method (Kabir et al., 2011). Eshghi compared traditional clustering methods and latent class models and found that using different clustering methods might produce different groups, suggesting that each methodology could lead to different interpretations (Eshghi et al., 2011). Ming Sun et al. (2019) utilised a latent class cluster model to segment pedestrians involved in collisions. Their findings implied that the latent cluster technique, when used in traffic safety studies, might uncover highly significant hidden factors. It also demonstrated the various benefits of latent class clustering over K-means clustering (Sun et al., 2019). Additionally, Yongwen Jiang et al. (2015) employed a latent class model to predict chronic disease patterns in cities and towns. The results showed that the latent class model distinguished three classes that reflected three levels of health indicators. It also suggested that using the latent class model was a very effective method for identifying patients to target chronic disease (Yongwen et al., 2015). Luzia Gonçalves et al. (2012) employed Bayesian latent class models to diagnose malaria, and their findings demonstrated that Bayesian latent class models were effective for this purpose (Gonçalves et al., 2012). As such, this chapter deals with MLC of a disease for which patients have multiple comorbidities. I explore the effectiveness of MLC for NB classifiers when a latent class model is used to cluster patients. The latent classes within the model can help to explain the relationships between the clusters and the comorbidities.

5.3 Latent class model

Latent class models are powerful improvements to traditional methods of clustering, factoring, regression, and neural network applications. Traditional methods express relationships among observed variables, whereas latent class models can include discrete unobserved variables. Latent class models have fewer biases because they do not rely on the model assumption. Latent class analysis (LCA) is a specific kind of latent class model. LCA (see chapter 3) is a hidden, discrete variable that divides cases into groups based on similar characteristics. This model clusters cases based on membership probabilities estimated directly from the model, where variables may be continuous, nominal, or ordinal (Magidson & Vermunt, 2005). In other words, LCA is a method that identifies hidden relationships among observed variables in order to cluster individuals into groups. LCA is a useful tool for improving health outcomes, as it can be used to find better characters that are unobservable within a population. It is a method that organises observed variables that represent unobservable clinical into meaningful subgroups. The resulting groups help clinical departments to diagnose diseases and personalise medicine (Law & Harrington, 2016). Utilising LCA in medicine allows for the production of crucial information. Likewise, group characteristics can be used to discover relevant latent subgroups that may result in distinct treatment outcomes. Advanced subgroup analyses provided by LCA in medicine aid in disease prevention and treatment (Lanza & Rhoades, 2013).

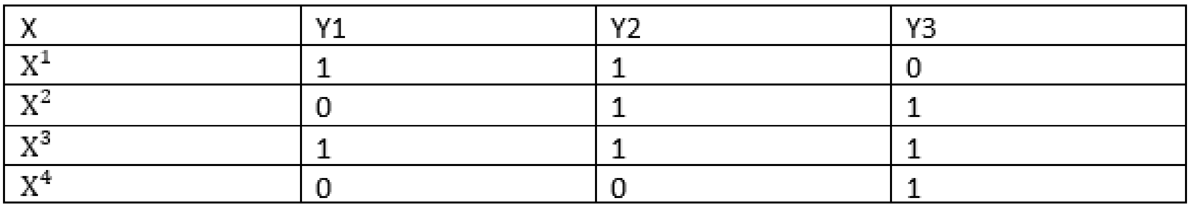
5.4 Multilabel classification

The difference between multi-class classification and MLC is that, in multi-class classification, each example belongs to only one class and is associated with only one label. By contrast, in MLC, each example can be associated with multiple labels. As an example of multi-class classification, a fruit can be an apple or a pear, but it cannot be both at the same time. As an example of MLC, a patient can have several symptoms at the same time (Bi & James, 2013). Single-label classification is a particular status of MLC in cases in which a patient has one disease. Multi-label classification methods have become an increasingly important subject in medicine due to the correlation of the labels (Ying et al., 2014). The aim of MLC is to predict these labels and determine whether there is a relationship among them. Correlations among these labels might lead to better decisions and predictions. However, there are two types of application methods that require MLC: problem transformation methods and algorithm adaptation methods. In addition, the evaluation metrics for MLC are slightly different than those for single label classification.

- **Transformation Methods**

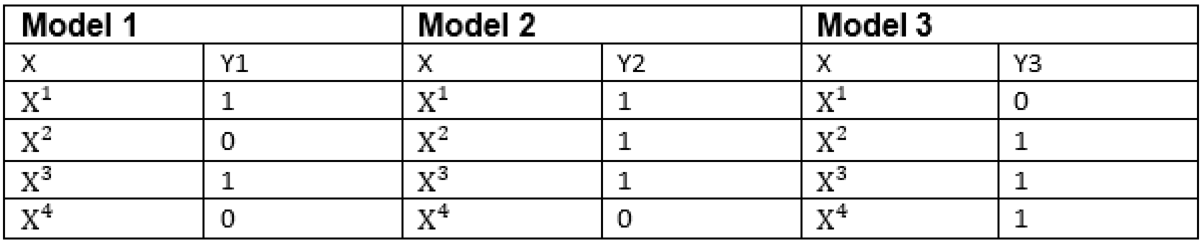
Transformation methods convert multilabel issues into a single label or multiple labels. Binary relevance (BR), label powerset (LP), and classifier chains (CC) methods are the main transformations methods used in MLC. BR is the most popular problem transformation method. This method predicts each binary label independently. The dataset is built for each binary label, and the output of an unseen instance is the union of the labels that have been predicted previously (Santos et al., 211). LP is a simple problem transformation method. It works by transforming a multilabel problem into a multi-class problem. It considers all labels that exist in the multilabel dataset as single-label classifications of multiple classes (Tsoumakas et al., 2009). Finally, the CC method, also called modelling label dependence, is based on BR. However, it overcomes BR's disadvantages and provides a better predictive performance. The chain is constructed by predicting each binary label independently. However, it considers the previous predictions of the binary label to be feature attributes. This algorithm considers the correlations among the labels (Read et al., 2011). Table 5.1 examines a dataset consisting of 4 cases and 4 labels (y1, y2, y3 and y4). Figures 5.1, 5.2, and 5.3 detail the implementation steps for each transformation method.

Table 5.1*: Example of Multilabel Dataset*



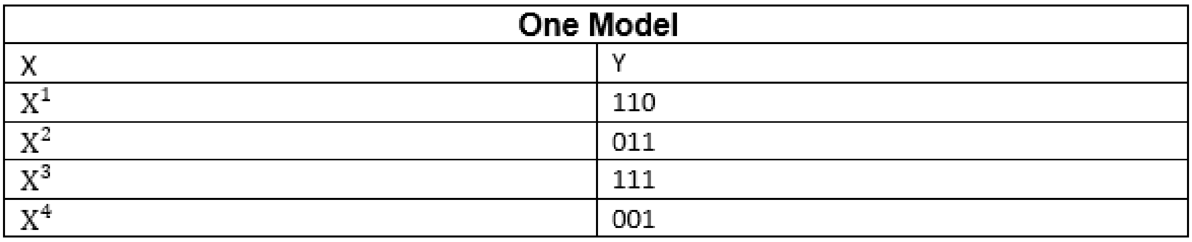
X	Y1	Y2	Y3
X^1	1	1	0
X^2	0	1	1
X^3	1	1	1
X^4	0	0	1

Table 5.2: *The Procedure of the BR Method for a Multilabel Dataset (Splitting the Dataset into Single Labels)*



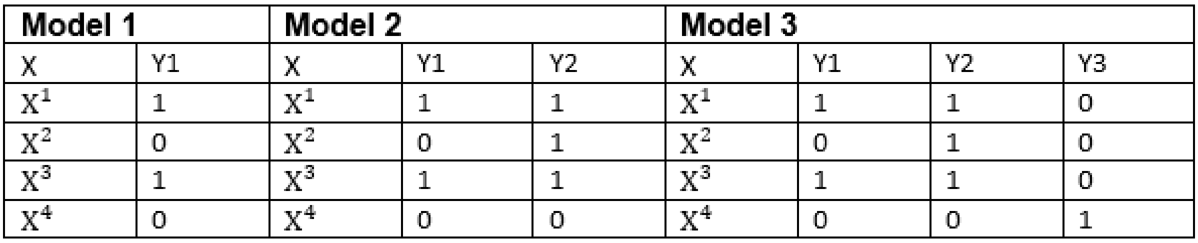
Model 1		**Model 2**		**Model 3**	
X	Y1	X	Y2	X	Y3
X^1	1	X^1	1	X^1	0
X^2	0	X^2	1	X^2	1
X^3	1	X^3	1	X^3	1
X^4	0	X^4	0	X^4	1

Table 5.3: *The Procedure of the LP method for a Multilabel Dataset (Converting the Dataset into a Single Class)*



One Model	
X	Y
X^1	110
X^2	011
X^3	111
X^4	001

Table 5.4*: The Procedure of the CC Method for a Multilabel Dataset (Converting the Dataset into Multiple Datasets, and Building an ML model in Each One)*



Model 1		**Model 2**			**Model 3**			
X	Y1	X	Y1	Y2	X	Y1	Y2	Y3
X^1	1	X^1	1	1	X^1	1	1	0
X^2	0	X^2	0	1	X^2	0	1	0
X^3	1	X^3	1	1	X^3	1	1	0
X^4	0	X^4	0	0	X^4	0	0	1

- **Algorithm Adaptation**

These methods are based on standard ML algorithms with structures modified to address multilabel problems. For instance, the multilabel DT algorithm was based on the DT algorithm. The C4.5 algorithm was adapted to be suitable for a multilabel dataset. They modified the entropy formula to address the multilabel problem; instead of calculating the entropy for one label, the new entropy formula collects all entropies for each label. Additionally, the multilabel k nearest neighbours (ML-KNN) algorithm is an extension of the KNN (K nearest neighbours) learning method using a Bayesian approach. It relies on prior and posterior probabilities for the frequency of each label (Aldrees et al., 2016). Thus, algorithm adaptation methods refer to the process of adapting ML methods to MLC problems.

- **Evaluation Metrics**

Multi label classification evaluation metrics assess the model by simultaneously considering the prediction of the labels. Thus, this model differs from a single learning model, in which model performance depends on each label independently. I investigated how the evaluation metrics for a MLC model are calculated (Tsoumakas & Katakis, 2009).

1- Accuracy

Accuracy is the ratio of correctly predicted labels to all labels for a given record. The average of all record accuracy is the overall accuracy.

$$Accuracy = \frac{1}{m}\sum_{i=1}^{m}\frac{|(Y_i \cap Z_i)|}{|(Y_i \cup Z_i)|} \qquad (5.1)$$

Y_i : the true labels, Z_i : the predicted labels, m: the number of instances.

2- Precision

Precision is the ratio of correctly predicted labels to all actual labels, averaged across all cases.

$$\Pr ecision = \frac{1}{m}\sum_{i=1}^{m}\frac{|(Y_i \cap Z_i)|}{|Z_i|} \qquad (5.2)$$

Y_i : the true labels, Z_i : the predicted labels, m: the number of instances.

3- Recall

Recall is the ratio of correctly predicted labels to all other predicted labels, averaged across all cases.

$$\text{Re}\,call = \frac{1}{m}\sum_{i=1}^{m}\frac{|(Y_i \cap Z_i)|}{|Y_i|} \tag{5.3}$$

Y_i : the true labels, Z_i : the predicted labels, m: the number of instances.

4- Hamming loss

Hamming loss is the ratio of incorrect labels to all labels. An approach is considered effective when the hamming loss value is low.

$$Ham\min g\_Loss = \frac{1}{m}\sum_{i=1}^{m}\frac{|(Y_i \Delta Z_i)|}{|L|} \tag{5.4}$$

Y_i : the true labels, Z_i : the predicted labels, m: the number of instances, L: the number of labels.

5.5 Latent class MLC naïve Bayes

In accordance with what was previously discussed, this chapter suggests a novel approach that addresses the problem of multiple comorbidities. This method combines the latent class and MLC models when applied to a dataset that measures multiple complications. This model first identifies hidden factors between observed data utilising a latent class variable that clusters patients into groups. This model then classifies each group using a naïve multilabel classifier to jointly predict their labels (Wei et al., 2011). Consequently, this algorithm comprises two phases that operate concurrently: LCA.

- **Latent Class Analysis**

According to my methodology, which was implemented using MATLAB software, the latent variable is considered a categorical variable with a number of classes. My method uses the EM algorithm to estimate the probabilities of these classes. This iterative algorithm follows two steps: the first is to estimate parameters, and the second is to update the model (Mooijaart & Heijden, 1992). My method starts by building a table containing all combinations of and frequencies between observed variables. Initially, I assigned a probability to each class of the latent variable based on a random number generator. The probability of joint observed variable was computed as follows:

$$P(y_1,....,y_m) = \sum_{c=1}^{C} P(L=c)P(y_1,....,y_m \setminus L=c) \tag{5.5}$$

$$P(y_1,....,y_m \setminus L=c) = \prod_{m=1}^{m} P(y_m \setminus L=c) \tag{5.6}$$

y: represents the observed variable for each and every possible combination of the observed variables, L: latent class variable, c: the class.

This procedure was repeated until the maximum value of the log-likelihood (LL) was reached, at which point I obtained the best possible latent class model.

$$LL = \sum_{c=1}^{L} \pi_c \, p_{c(y)} \tag{5.7}$$

y: represents the observed variable for each and every possible combination of the observed variables, L: latent class variable, c: the class.

Additionally, in order to decide how many classes to assign to the latent variable, I used Bayesian information criteria (BIC). The number of classes was decided to the lowest value of BIC:

$$BIC = -2 * LL + P * Ln(n) \tag{5.8}$$

where *n* is the sample size, and *P* is the number of parameters, LL is the log-likelihood

Finally, I assigned each patient to the class to which they most likely belonged in order to cluster patients within subgroups.

- **MLC Naïve Bayes**

After applying a latent class model to cluster patients into subgroups, I constructed an adaptive NB MLC classifier for each subgroup in order to predict all labels for unseen patients. The NB method was adapted to deal with conditional relationships among the different labels. The aim of this method was to predict the values of all labels and discover whether the use of this, compared to multiple single class models, improves classification performance (by assuming conditional relationships among the different labels). The conditional probability of patient pi with relate to each class label p_i is defined as follows:

$$P(l_j \backslash p_i) = \frac{P(l_j)P(p_i \backslash l_j)}{P(p_i)} \quad (5.9)$$

$$P(L = l_i) = \frac{N_j}{N} \quad (5.10)$$

where N_j is the number of values having the label l_j.

$$P(a_k \backslash l_j) = \frac{1 + N_{kj}}{m + \sum_{k=1}^{m} N_{kj}} \quad (5.11)$$

where N_{kj} is the total frequency with which values a_k appear in individual cases in category l_j.

When predicting patients who haven't yet been seen, NB was constructed based on the aforementioned parameters; label relationships were also taken into consideration. Using the conditional probability between the label and the features, I forecast the initial label. Additionally, I used the conditional probability between another label and the features, as well as the conditional probability between this label and the first label, to predict it. Thus, I calculated the average posterior probability of patient p_i in each class as follows:

$$P_{average} = \frac{1}{n} \sum_{j=1}^{n} P(l_{j \backslash} p_i) \quad (5.12)$$

Thus, after calculating the label values for cases in which the class was positive, I calculated the average posterior probability. If the conditional probability for label *j* was greater than or equal to the mean posterior probability, the value of label *j* was considered positive; otherwise, it was not. I repeated the same procedure for cases in which the class was negative. I returned the values of labels with the highest posterior probability. As a result, my suggested method combines LCA, which clusters patients into groups, with the NB MLC classifier in order to predict multiple labels, as NB has been shown to be an effective classifier in multilabel learning.

The following pseudocode explains the steps used in order to build the new algorithm.

Algorithm 2 Pseudocode of Latent Naïve Bayes Multi Label Classification

```
Input: Dataset of Patient Features and Labels.
Output: Clusters of patients and a multilabel NB model for each group.
Begin
For $i = 1$ to 10
    Using 10-fold cross validation, divide the data into 90% training data and 10% test data.
    Build latent class model for the training data using EM algorithm.
    Output LC (patient groups).
    Build the multi-label naïve Bayes model for each group.
End for
For $j = 1$ to test data
    Assign each test data example to one of the above groups using the scoring formula (membership formula).
    Compute the conditional probability and features for each label.
    Compute the average posterior probability.
End for
Compute accuracy and other metrics.
End
```

The following flowchart outlines the basic steps of my proposed method.

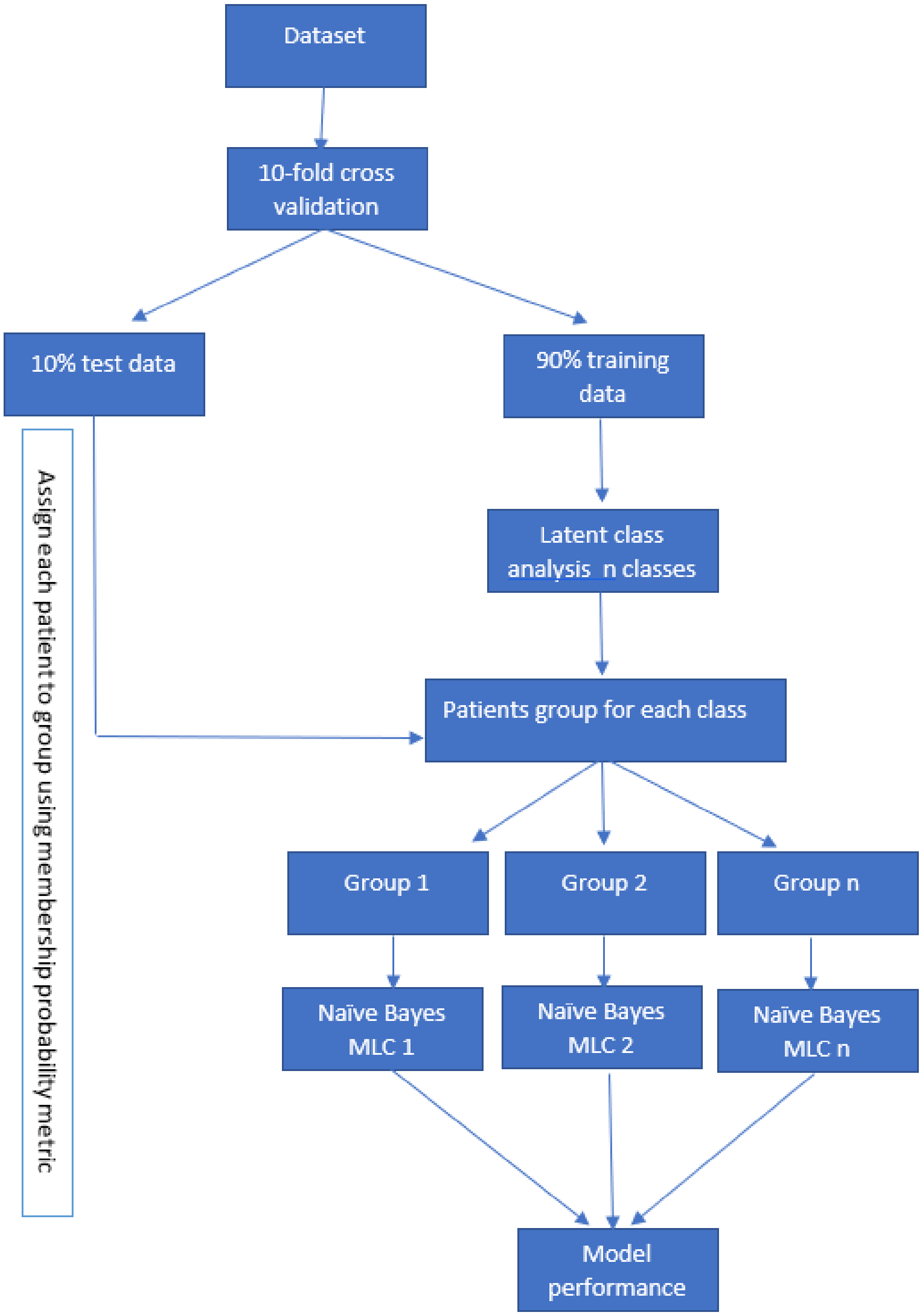


Figure 5.1: *Flowchart of Latent Class MLC Naïve Bayes Method*

5.6 Datasets

I explore two datasets in this chapter: SSc, and coronary heart disease in traditional Chinese medicine.

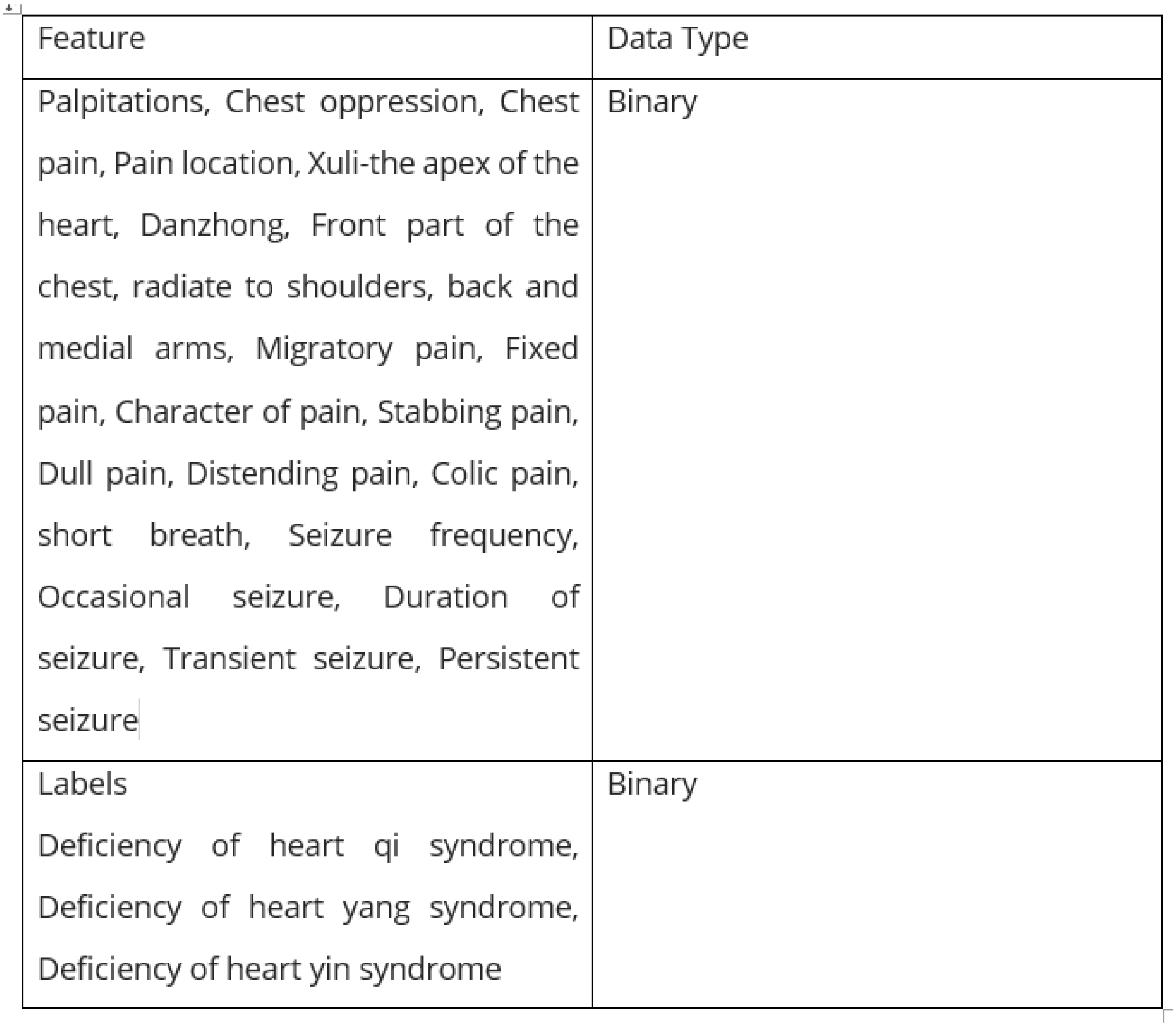
SSc Dataset

This dataset was provided by the Royal London Hospital (see chapter 3), and I used it to implement my new method (latent class MLC NB). This dataset contained 677 patients.

Coronary Heart Disease

This dataset was collected by Shanghai University, and it was designed for multilabel learning tasks. The data contained 555 patients, 265 men and 290 women. The names of the features and their data types are listed in the table below. The experiments performed on the aforementioned datasets using the aforementioned proposed method are described in the following section.

Table 5.5: *Coronary Heart Disease Variables*



Feature	Data Type
Palpitations, Chest oppression, Chest pain, Pain location, Xuli-the apex of the heart, Danzhong, Front part of the chest, radiate to shoulders, back and medial arms, Migratory pain, Fixed pain, Character of pain, Stabbing pain, Dull pain, Distending pain, Colic pain, short breath, Seizure frequency, Occasional seizure, Duration of seizure, Transient seizure, Persistent seizure	Binary
Labels Deficiency of heart qi syndrome, Deficiency of heart yang syndrome, Deficiency of heart yin syndrome	Binary

5.7 Experiments

This section explores experiments performed to execute my proposed method using the SSc dataset and the coronary heart disease dataset. The proposed method attempts to jointly predict T2RIP, time to develop pulmonary fibrosis (PF), and time to develop pulmonary hypertension (PAH; see chapter 3). I selected all patients from this dataset who developed at least one of the above classes within the first 5 years and all patients who did not develop at least one within 5 years. The predicted classes have two values: ‘1’, representing patients who could have an event before 5 years, and ‘2’, representing patients who could have an event after 5 years. The suggested method was used to predict all labels of the previously described coronary heart disease dataset.

The purpose of this chapter is to describe the implementation of the prosed model, analyse it, and compare it to other common models. I also attempt to contrast my model with further MLC techniques. Based on my experiences and the proposed method described in the preceding section, I thus investigated the following:

1- Using a standard single-label NB model to predict classes as well as with LCA model.
2- Using a multilabel NB classifier to predict classes.
3- Using the standard multilabel transformation methods BR and CC, which were applied using standard logistic regression (LR) and SVM, to predict classes.
4- Using my proposed method, latent class NB MLC, to predict classes.

All of the above have been used to investigate the SSc dataset to predict T2RIP, time to develop PAH, and time to develop PF. Additionally, I performed the following analyses:

1- Single-label sensitivity analyses.
2- Multi-label sensitivity analyses.
3- Analysis of the optimal number of clusters.
4- Comparison of the proposed method to other MLC methods.
5- Data analysis of the discovered groups within the clinical context.

The results of these experiments, which I carried out using MATALAB software, are examined in the next section.

5.8 Results

After describing the suggested method and the experiments I performed for this study, I describe the results of my experiments in this section. Boxplots, a straightforward graphical tool, are used to interpret, compare, and communicate my results. This shape, which I use to display the error rate in my research, also displays the minimum value, maximum value, and median of a set of data. Additionally, the significance difference between two means was compared using t-tests. In addition to using single-label evaluation metrics, I employed multilabel evaluation measures to evaluate my model and compare it to other MLC methods.

A) Systemic sclerosis

I ran the standard NB model, the multilabel NB (MLNB) classification model, the standard NB with latent class (LCNB) model, and multilabel NB classification with latent class (LCMLNB) model in order to predict T2RIP, time to develop PF, and time to develop PH. The following plots display the results of these methods on all of the test data. They show that the MLC with latent class model performed better than the standard NB and MLC models. My method, LCMLNB, performed better than all other algorithms. It performed significantly better than standard NB, with p equal to 0.035 (< 0.05) for T2RIP, p equal to 0.022 (< 0.05) for time to develop PF, and p equal to 0.037 (< 0.05) for time to develop PAH. These results demonstrate that the error rate of the classifier decreases when patients are sorted into groups with comparable characteristics, as opposed to supervised learning methods. In addition, results indicate that the error rate of the classifier decreases when the relationships among the labels are considered.

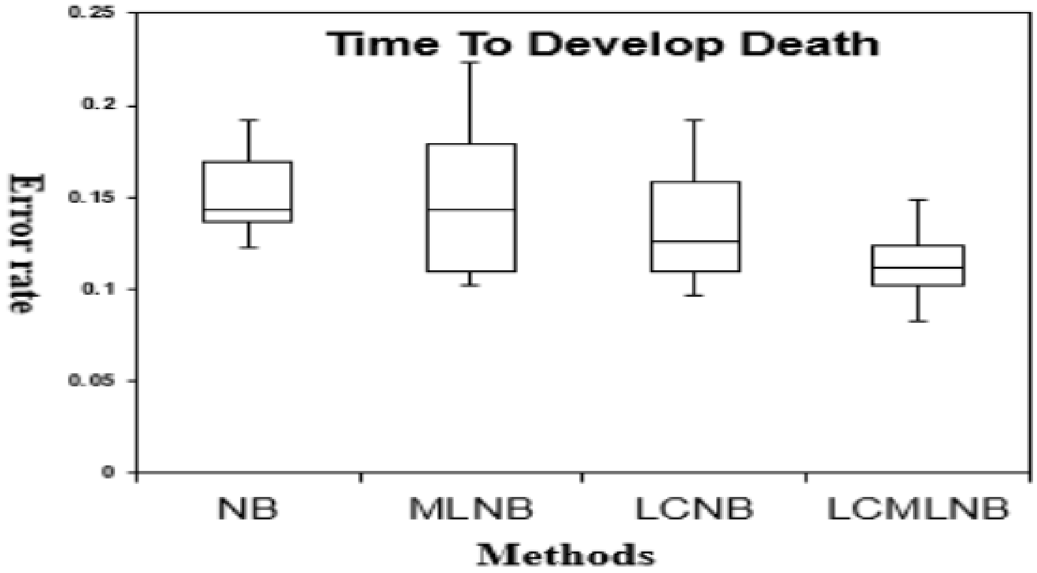


Figure 5.2*: Comparison Between Latent Class Multi Label Classification Model With Other Methods to Predict Time to Death*

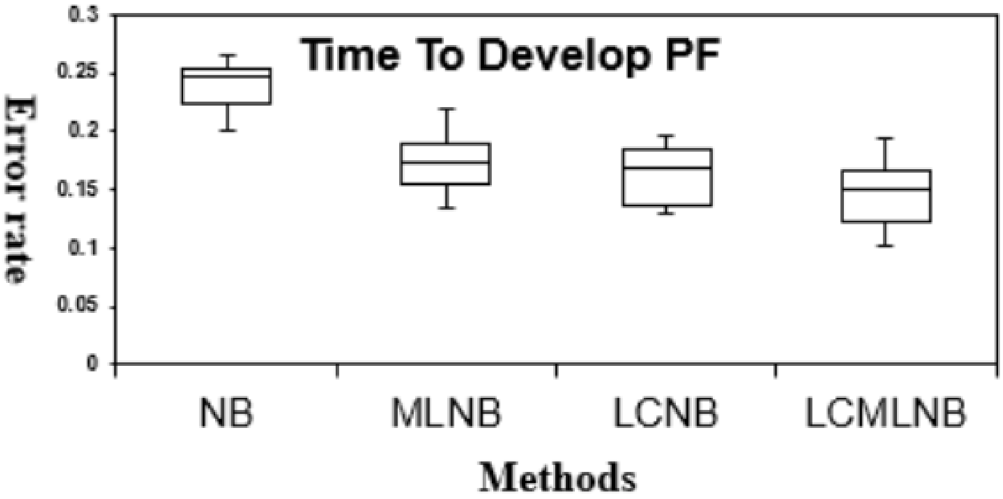


***Figure 5.3**: Comparison Between Latent Class Multi Label Classification Model With Other Methods to Predict Time to Develop PF*

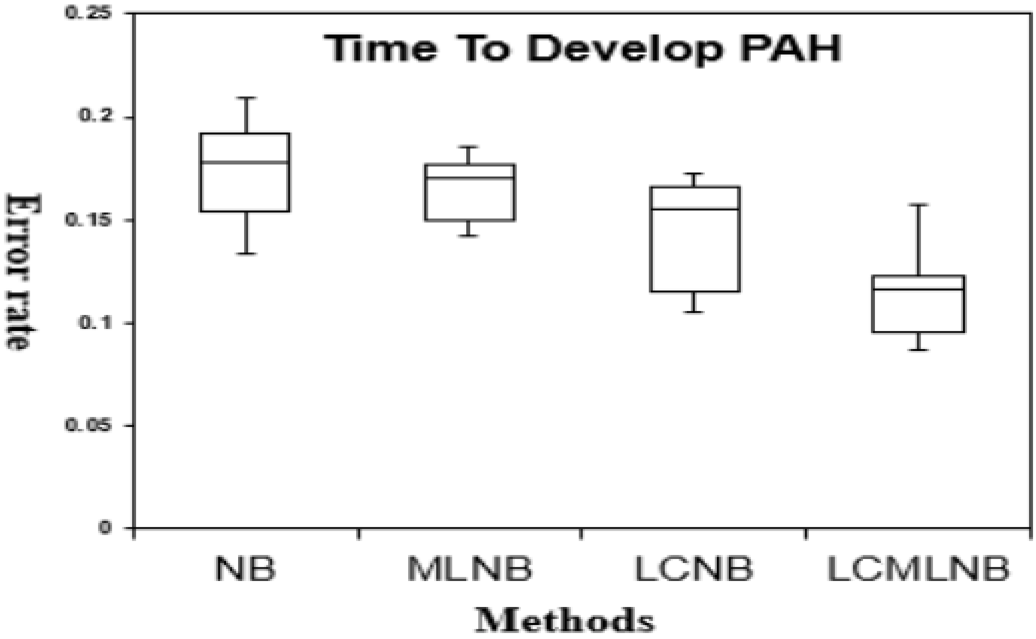
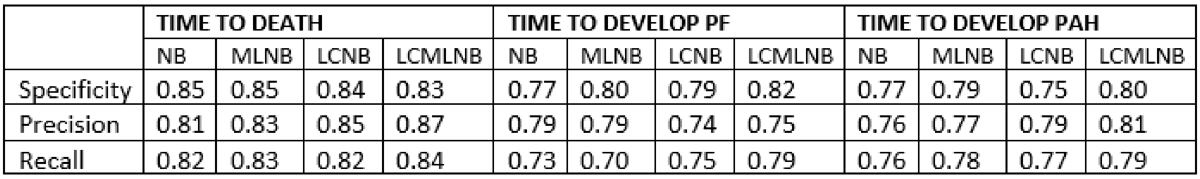


***Fig 5.4**: Comparison Between Latent Class Multi Label Classification Model With Other Methods to Predict Time to Develop PAH*

Additionally, in order to demonstrate whether my method predicted the classes of SSc more successfully than other methods, I computed performance metrics, as shown in Table 5.3. The results showed that my proposed method, LCMLNB, improved performance. In other words, LCMLNB improved prediction of which patients could die, develop PF, or develop PAH during the first 5 years compared to other methods, as measured by the recall metric. This could be due to the fact that my method successfully clustered patients into distinct groups, as well as the fact that the relationships among the labels affect performance metrics. These results demonstrate that the use of a latent class model that clusters patients into groups while also accounting for the relationships among the predicted classes enhances the model's performance.

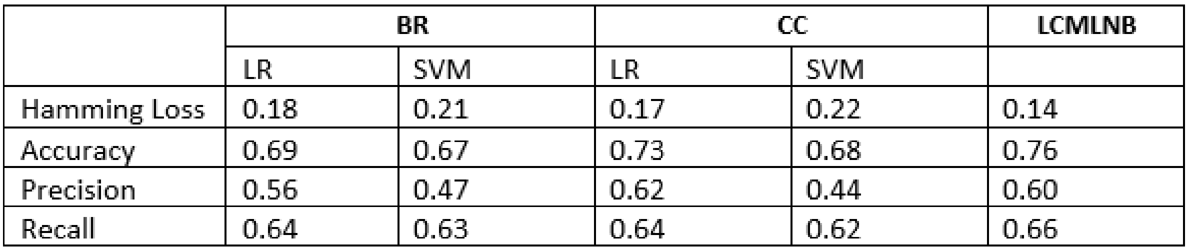
Table 5.6: *Metrics Measures Results for NB, MLNB, LCNB, and LCMLNB for Time to Death Class, Time to Develop PAH Class, and Time to Develop PF Class*



	TIME TO DEATH				TIME TO DEVELOP PF				TIME TO DEVELOP PAH			
	NB	MLNB	LCNB	LCMLNB	NB	MLNB	LCNB	LCMLNB	NB	MLNB	LCNB	LCMLNB
Specificity	0.85	0.85	0.84	0.83	0.77	0.80	0.79	0.82	0.77	0.79	0.75	0.80
Precision	0.81	0.83	0.85	0.87	0.79	0.79	0.74	0.75	0.76	0.77	0.79	0.81
Recall	0.82	0.83	0.82	0.84	0.73	0.70	0.75	0.79	0.76	0.78	0.77	0.79

The aforementioned outcomes demonstrate how well my model performed for each single label separately, but in order to assess a multilabel classifier, I had to employ other metrics (described previously for multilabel classifier) that were assessed using each example (i.e. per patient for all labels predicted together). In order to employ these metrics, I also implemented other multilabel classifiers that were based on transformation methods. I applied BR and CC approaches using standard LR and SVM to predict classes. As stated previously, implementing these approaches is straightforward. The BR approach seeks to divide the dataset into numerous datasets, each of which has a unique label. Standard LR and SVM were applied to each sub-dataset to predict every label, and the results were the union of all labels. By contrast, CC follows an identical procedure, except each sub-dataset uses the previous labels as features to predict the current label. Table 5.4 outlines the multilabel performance metrics results for my proposed model as compared to other multilabel classifiers. It is evident that CC classifiers outperformed BR classifiers; this is because CC classifiers take label dependencies into account. However, my model outperformed CC classifiers because it considers label dependencies and the fact that patients were clustered into distinct groups.

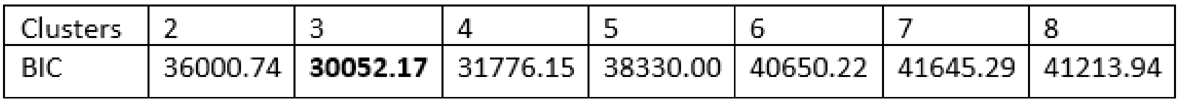
***Table 5.7**: Metrics Measures Results for BR-Logistic Regression, BR-SVM, CC-Logistic Regression, CC-SVM, and LCMLNB for MLC Time to Death Class, Time to Develop PAH Class, and Time to Develop PF Class*



	BR		CC		LCMLNB
	LR	SVM	LR	SVM	
Hamming Loss	0.18	0.21	0.17	0.22	0.14
Accuracy	0.69	0.67	0.73	0.68	0.76
Precision	0.56	0.47	0.62	0.44	0.60
Recall	0.64	0.63	0.64	0.62	0.66

Additionally, it is necessary to discover the meaning of the latent class model in my dataset. In my model, the latent class model split the dataset into three groups. All patients within the same group have comparable personalities. This dataset was separated into three groups due to the fact that the BIC (see section 5) was lowest when the dataset was divided into three groups. The following table displays the BIC values for several scenarios in which the dataset was partitioned into 3, 4, 5, 6, 7, or 8 clusters.

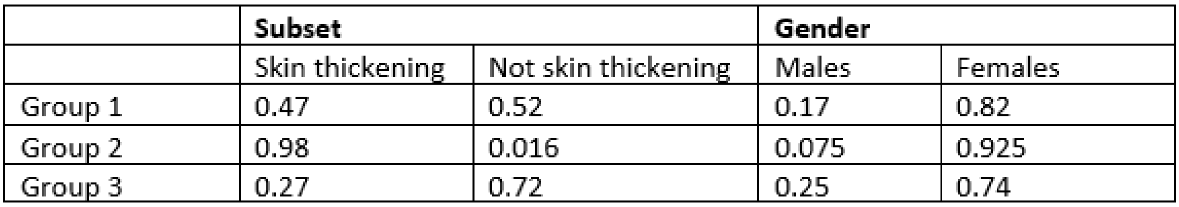
***Table 5.8:** Bayesian Information Criteria (BIC) Values for Different Clusters in SS Dataset*



Clusters	2	3	4	5	6	7	8
BIC	36000.74	**30052.17**	31776.15	38330.00	40650.22	41645.29	41213.94

The patients were separated into three groups with cluster sizes of 47%, 31%, and 22%. The following tables show the percentage difference for blood test results and antibody information among group members. They show that most of the patients in group 2 experienced skin thickening and were female. It can be seen from the results that Hb levels were lower in group 3 than in the other groups, while Cr levels were higher. Finally, regarding lung function, results showed that FVC and DLCO were higher in group 2 than in the other groups. All these results can help clinicians to better identify individual patients’ characteristics in order to personalise their care plans whilst improving prediction of long-term outcomes.

***Table 5.9**: Subset and Gender Variables Distribution as a Percentage Within Groups*



	Subset		**Gender**	
	Skin thickening	Not skin thickening	Males	Females
Group 1	0.47	0.52	0.17	0.82
Group 2	0.98	0.016	0.075	0.925
Group 3	0.27	0.72	0.25	0.74

Table 5.10: *Distribution of Hb and Cr Variables as Values Within Groups*

	Hb	Cr
Group 1	12.83	74.37
Group 2	12.98	76.89
Group 3	**10.87**	**257.59**

The following graph shows that the percentage of group 1 patients who developed both PAH and PF and died within 5 years was higher than for the other groups. Additionally, the percentage of patients in groups 2 and 3 who developed PF was low compared to those who died or developed PAH.

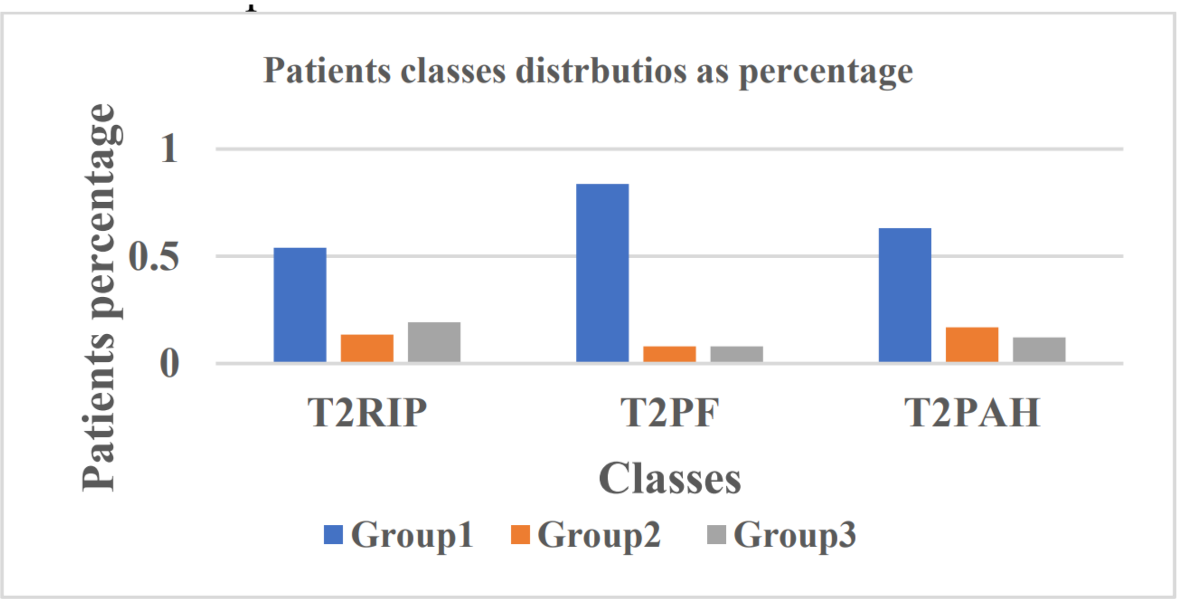
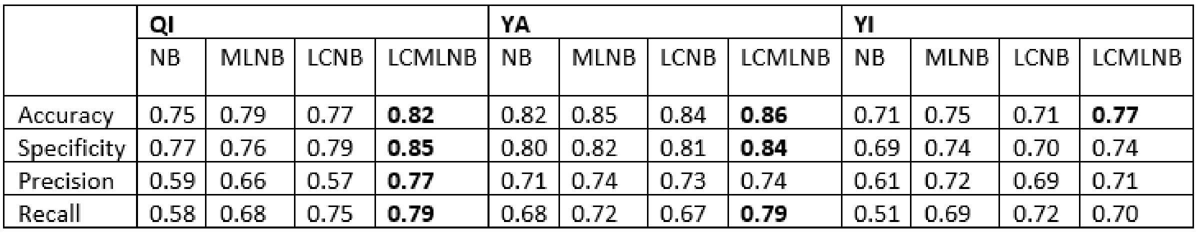


***Figure 5.5**: Patients Classes Distributions Within the Groups*

B) Coronary heart disease

I re-applied the standard NB model, the MLNB classification model, the LCNB model, and the LCMLNB model in order to predict coronary heart disease labels (see Table 5.2). This dataset was designed by Guo-Ping et al. for use in multilabel learning tasks (Liu et al., 2010). I used this dataset to validate my proposed method. The resulting performance metrics of my proposed method are shown in Table 5.5, along with comparisons to other approaches to predicting deficiencies of heart qi syndrome (QI), heart yang syndrome (YA), and heart yin syndrome (YI). These results were computed separately for each label. When a multilabel classifier and latent class model were taken into account, the performance metrics per label generally increased, as shown in Table 5.8.

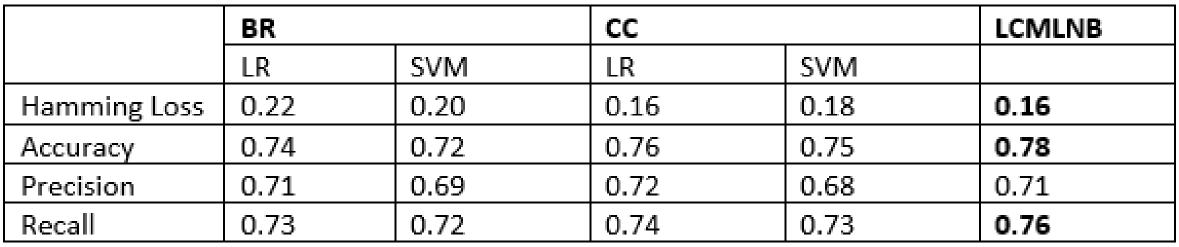
***Table 5.11**. Metrics Measures Results for NB, MLNB, LCNB, and LCMLNB for QI, YA, and YI Classes*



	QI				**YA**				**YI**			
	NB	MLNB	LCNB	LCMLNB	NB	MLNB	LCNB	LCMLNB	NB	MLNB	LCNB	LCMLNB
Accuracy	0.75	0.79	0.77	**0.82**	0.82	0.85	0.84	**0.86**	0.71	0.75	0.71	**0.77**
Specificity	0.77	0.76	0.79	**0.85**	0.80	0.82	0.81	**0.84**	0.69	0.74	0.70	0.74
Precision	0.59	0.66	0.57	**0.77**	0.71	0.74	0.73	0.74	0.61	0.72	0.69	0.71
Recall	0.58	0.68	0.75	**0.79**	0.68	0.72	0.67	**0.79**	0.51	0.69	0.72	0.70

Again, the aforementioned results illustrate how well my model worked for each single label separately. However, in order to evaluate a multilabel classifier, I utilised additional metrics that were evaluated using each case (i.e. per patient for all labels predicted together). As in the SSc dataset, I implemented BR and CC approaches, which were applied using standard LR and SVM, to predict these labels together. Table 5.9 outlines the multilabel performance metrics results for my proposed model in comparison to other multilabel classifiers. The outcomes demonstrate improvement from one classifier to the next. Additionally, the outcomes demonstrate that my suggested strategy outperforms other methods; additionally, it is a transparent model that could offer clinicians useful information when patients are clustered.

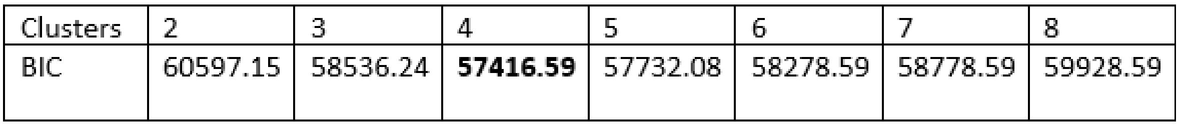
***Table 5.12.** Metrics Measures Results for BR-Logistic Regression, BR-SVM, CC-Logistic Regression, CC-SVM, and LCMLNB for MLC QI, YA, and YI Classes Together*



	BR		**CC**		**LCMLNB**
	LR	SVM	LR	SVM	
Hamming Loss	0.22	0.20	0.16	0.18	**0.16**
Accuracy	0.74	0.72	0.76	0.75	**0.78**
Precision	0.71	0.69	0.72	0.68	0.71
Recall	0.73	0.72	0.74	0.73	**0.76**

As for the SSc dataset, it was necessary to determine the meaning of the latent class model in my dataset. The latent class model in my model split the dataset into four groups. All patients from the same group had comparable personalities. This dataset was separated into four groups due to the fact that the BIC was lowest when the dataset was divided into four groups. Table 5.10 presents the BIC value for several clusters.

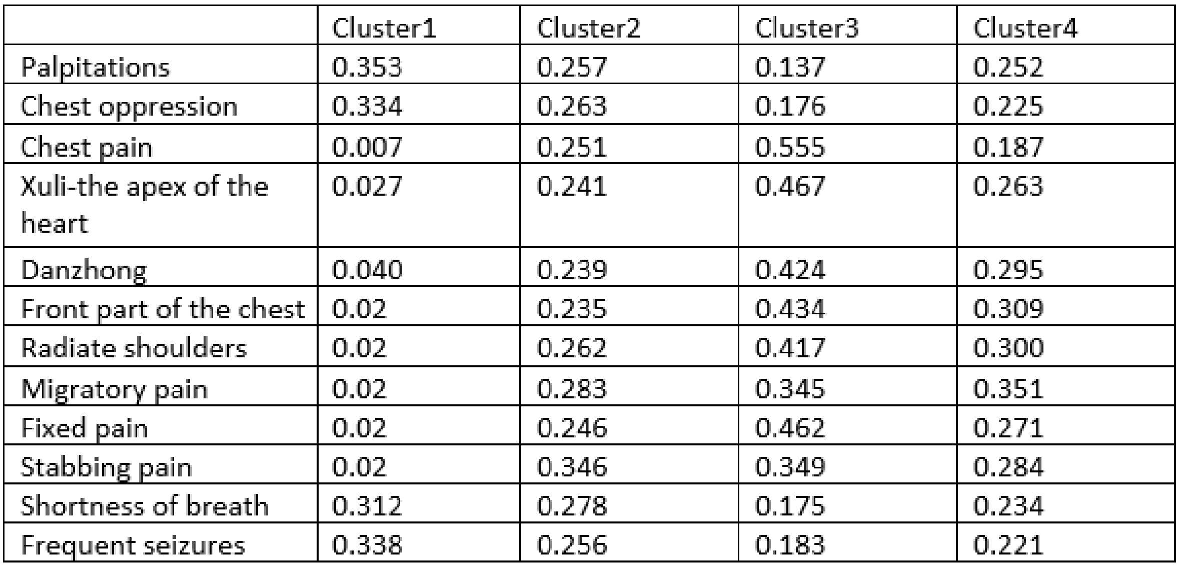
***Table 5.13:** Bayesian Information Criteria (BIC) Values for Different Clusters in Coronary Heart Disease Dataset*



Clusters	2	3	4	5	6	7	8
BIC	60597.15	58536.24	**57416.59**	57732.08	58278.59	58778.59	59928.59

The patients were separated into three groups with cluster sizes of 45%, 25%, 21%, and 9%. The following tables show the percentage of occurrence of the symptoms within the groups. They demonstrate that group 1 had nearly no individuals with the symptoms chest front, migraine pain, fixed discomfort, or stabbing pain. Additionally, over half of the individuals with chest pain belonged to group 3. All of these discoveries can help doctors better comprehend a patient's condition and improve the care they provide.

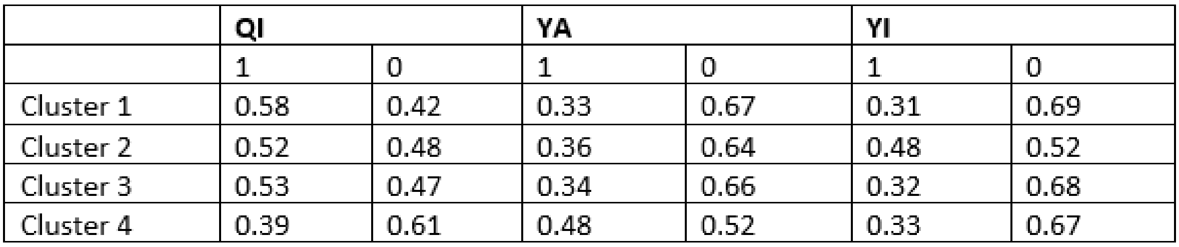
***Table 5.14:** Percentage of Occurrence of the Symptoms Within the Groups*



	Cluster1	Cluster2	Cluster3	Cluster4
Palpitations	0.353	0.257	0.137	0.252
Chest oppression	0.334	0.263	0.176	0.225
Chest pain	0.007	0.251	0.555	0.187
Xuli-the apex of the heart	0.027	0.241	0.467	0.263
Danzhong	0.040	0.239	0.424	0.295
Front part of the chest	0.02	0.235	0.434	0.309
Radiate shoulders	0.02	0.262	0.417	0.300
Migratory pain	0.02	0.283	0.345	0.351
Fixed pain	0.02	0.246	0.462	0.271
Stabbing pain	0.02	0.346	0.349	0.284
Shortness of breath	0.312	0.278	0.175	0.234
Frequent seizures	0.338	0.256	0.183	0.221

Additionally, the following table displays the percentage of occurrence/non-occurrence of the labels for each group. It demonstrates that patients in group 4 were less likely to experience QI than patients who did not experience QI. Thus, clinics could find this information helpful for individualised diagnosis and treatment.

***Table 5.15:** Percentage of Occurrence/Non-occurrence of the Labels (QI, YA, and YI) for Each Group*



	QI		**YA**		**YI**	
	1	0	1	0	1	0
Cluster 1	0.58	0.42	0.33	0.67	0.31	0.69
Cluster 2	0.52	0.48	0.36	0.64	0.48	0.52
Cluster 3	0.53	0.47	0.34	0.66	0.32	0.68
Cluster 4	0.39	0.61	0.48	0.52	0.33	0.67

5.9 Conclusion

In this chapter, I proposed an ML model that can handle multiple labels that may arise in real-world cases, where patients may have multiple diseases at once. This model utilises the power of LCA to find hidden factors by dividing patients into groups. For each group, the adaptive MLC NB model was used to predict all labels simultaneously. Therefore, this chapter first discussed the significance of MLC and LCA models in healthcare. It also examined the definition of MLC and the approaches utilised to address this issue as well as the definition of LCA. Second, this chapter described the proposed method, the experiments I conducted, and the datasets I used. Finally, this chapter concluded with a presentation and interpretation of the outcomes of my experiments. The results showed that my model outperformed the other methods; additionally, it is a transparent model that could aid in providing individualised clinical care. My approach uses a naïve Bayes algorithm adaptation method to assist in the prediction of multiple diseases at the same time. Thus, it is essential to address new machine learning algorithm adaptation methods that deal with MLC issues. My techniques in Chapters 4 and 5 addressed sub-disease identification by grouping patients into clusters and constructing supervised machine learning in each group. These methods have been used with various healthcare datasets, and it has been demonstrated that clustering patients into robust groups improves model performance and aids in developing personalised medicine. However, AI models may become obsolete over time, resulting in a considerable performance decline known as concept drift. Consequently, it is vital to address the issue of concept drift and the approaches that may be used to monitor and identify concept drift in AI models. In the next chapter, I introduce the problem of concept drift as well as the methods that can be used to monitor machine learning algorithms over time. These strategies will avoid concept drift by continuously checking the ML model and updating it when drift is observed.

Chapter 6

Concept Drift in Healthcare

6.1 Introduction

As artificial intelligence (AI) and particularly ML have matured as disciplines, the availability of large medical datasets means that algorithms are being designed that can model complex medical domains and offer great potential to clinical healthcare. However, utilising models trained solely on historical data carries significant hazards, as they may become obsolete over time owing to concept drift. Therefore, this chapter first examines the significance of concept drift and its effects on healthcare AI models. It describes the concept drift phenomenon, its different types, and the detectors that can be used to address this issue. Second, this chapter describes the proposed updated concept drift method (DDM) employed in my research to detect concept drift as well as the datasets and experiments I implemented. Finally, the findings of the application of the DDM approach to the simulated, SSc datasets, and synthetic COVID-19 are covered and interpreted in this chapter.

6.2 Overview

The healthcare sector has already incorporated ML technologies to reap the enormous benefits of modern science and technology. Medical and surgical computers are being taught, with the aid of advanced supervised learning and ML models, to flawlessly conduct a wide range of medical procedures, including diagnostic, therapeutic, and surgical procedures. However, the growing application and adoption of ML models in the health sciences might create the risk of concept drift over time due to changes in the characteristics of presenting patients. Additionally, this drift may be gradual and caused by factors such as modifications to national health policy, or it may be abrupt and caused by factors including those associated with the COVID-19 pandemic (Duckworth et al., 2021). In their study, Beyene et al. (2015) emphasised the issue of prediction models declining in performance due to changing circumstances, such as when a patient requiring surgery is transferred from primary to secondary care. They hypothesised that effectively managing concept drift could improve surgical prediction accuracy and that this could be done using concept drift handling methods (Beyene et al., 2015). According to Hamish Huggard et al. (2020), changes in policy, financing, staff, or other factors could impact the performance of the ML model, which was trained to predict triage of care patients' decisions. To address this problem, they developed a new drift detection algorithm that monitors ML performance, which could be useful in any area (Hamish et al.,2020). However, provided a method for dynamically presenting sequential data features that aims to enhance the understanding of concept drift, which leads to a significant decline in classification performance. Thus, in medical applications, the concept drift problem, which degrades ML model performance over time, must be addressed regularly in order to ensure that predictions and diagnoses are safe and accurate.

6.3 Concept drift definition and types

Typically, ML builds static models using historical data. However, these models may become unreliable over time because real-world applications undergo continual changes (Webb et al., 2017). In other words, concept drift refers to hidden changes in the relationship between the input and output data that affects the performance of supervised learning models. Concept drift occurs when the data distribution changes due to a non-stationary environment. Drift can be either *real* drift (concept drift) or *virtual* drift. Real concept drift means that the conditional distribution of the target variable (i.e. prediction variable) changes without changing the distribution of the input data. A real-world example of concept drift is a change in human behaviour. ML models that predict hospitalisation, for example, could be ruined by a real-world change, such as the COVID-19 pandemic. Virtual drift, by contrast, occurs when the distribution of input data changes without affecting the target variable (Gama et al., 2014). The following image depicts real concept drift and virtual drift. The first image shows how the original data is distributed. The target variable decision boundary changes during actual concept drift but not virtual concept drift.

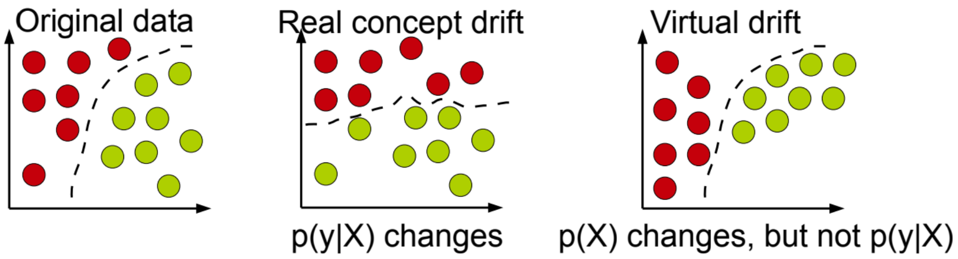


Figure 6.1: *Types of Drift (Gama et al., 2014)*

Lu and colleagues (2018) clarified that concept drift can be sudden, gradual, incremental, or reoccurring. When the performance of a ML model drops dramatically in a short period of time, this is known as sudden drift. The COVID-19 pandemic, for example, resulted in a significant drift in consumer behaviour within a very short time. It takes a considerable amount of time for incremental and gradual drift to fully result in the emergence of a new concept. The requirements for offering a loan, for instance, change gradually over time. Reoccurring drift happens when ML models are accurate at some times but not others. Some ML models, for example, are accurate throughout the summer but not during the winter. The following figure explains concept drift changes over time (Lu et al., 2018).

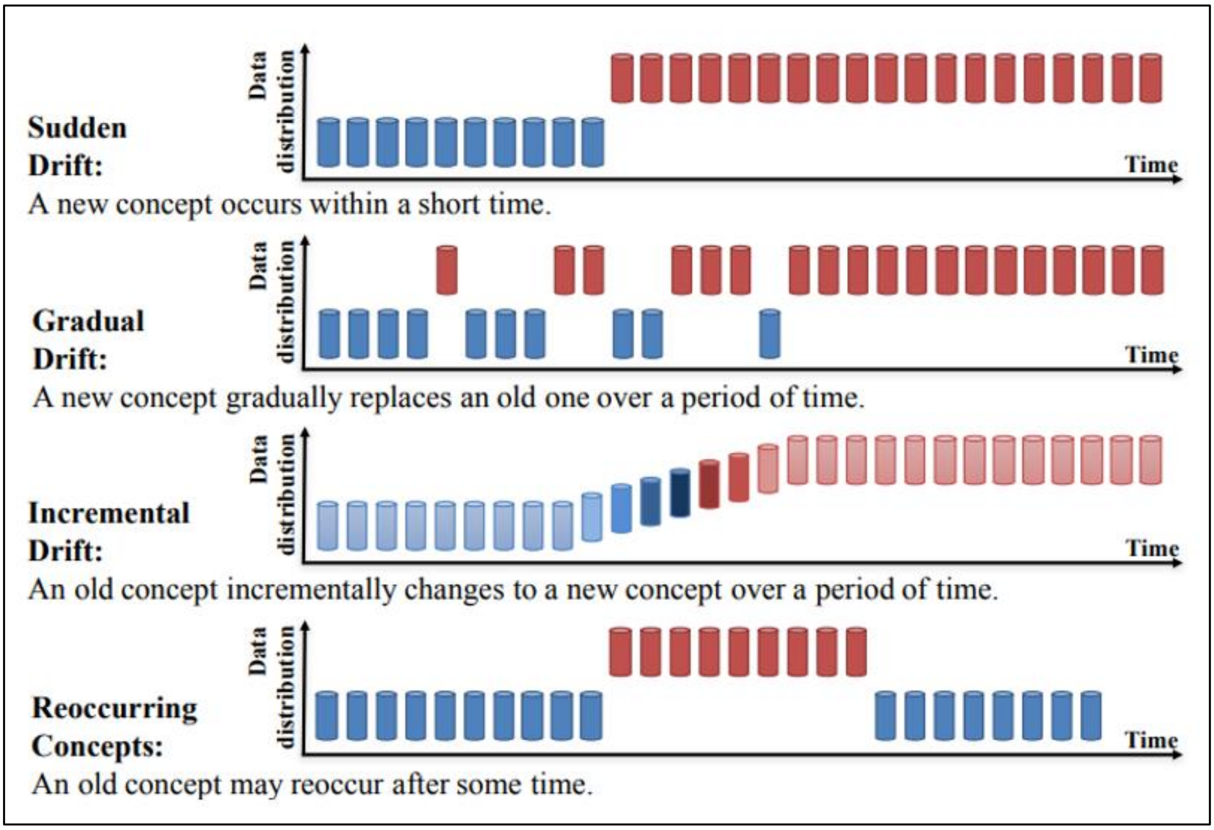

Sudden Drift:
A new concept occurs within a short time.
Gradual Drift:
A new concept gradually replaces an old one over a period of time.
Incremental Drift:
An old concept incrementally changes to a new concept over a period of time.
Reoccurring Concepts:
An old concept may reoccur after some time.
Data distribution
Time

***Figure 6.2:** Concept Drift Changes Over Time (Lu et al., 2018)*

As I examined the problem of concept drift, which affects ML models, I found that the biggest challenge is preventing its occurrence. One way to address concept drift is to build an initial ML model using historical data and update the model only when concept drift occurs. This approach uses concept drift detectors such as DDM, early drift detection method (EDDM), and ADWIN to monitor model performance and act in the event of concept drift. However, there are approaches for updating the model by adding new data to it on a regular basis (e.g. monthly or annually). There are also techniques for weighing the importance of input data, with recent data taking precedence.

6.4 Concept drift detectors

A variety of strategies can be used to detect concept drift. However, there are a few popular algorithms that I highlight in my research.

- **Drift Detection Method (DDM)**

The drift detection method (DDM) is one of the concept drift detection approaches that has already been established to detect concept drift. Gama et al. (2014) introduced this approach for dealing with streaming data. As long as data distribution is stationary, this approach assumes that the overall error rate will drop or remain constant for incoming instances. It employs a base learner to categorise incoming instances, and the classification result is used to calculate the base learner's online error rate. A supervised ML model predicts the class of each incoming instance and compares the predicted class value to the actual class value as it becomes available. Basically, the classification model shows whether or not the incoming instance was predicted correctly. If the incoming instance was predicted accurately, the overall error rate decreases, whereas if it was predicted inaccurately, the overall error rate increases. Consequently, a significant increase in the error rate signals concept drift. For this approach to be implemented, the error rate and standard deviation for each incoming instance must be computed. When the error rate and standard deviation reach a certain level (i.e. alert level), this signals the possibility of future concept drift. When the error rate and standard deviation reach a certain higher level (i.e. drift level), this indicates that concept drift has occurred. This method is effective for detecting both abrupt and gradual changes (Gonçalves et al., 2014). I utilised this technique in my research to handle batches of data, as opposed to streaming data, because it is a well-established algorithm for detecting concept drift.

- **Early Drift Detection Method (EDDM)**

Baena-Garca et al. (2006) created the EDDM approach to improve the identification of gradual concept drift. This algorithm works in the same way as DDM. However, instead of calculating the overall error rate of each incoming instance, it computes the average distance between two errors. Put differently, this technique calculates the distance between two instances that were incorrectly predicted. Thus, error rate and standard deviation must be computed. When the average error rate and standard deviation reach a certain level (i.e. alert level), this signals the possibility of future concept drift. When the error rate and standard deviation reach a higher certain level (i.e. drift level), this indicates that concept drift has occurred. This method takes concept drift into account when at least 30 errors have occurred. This strategy allows for the early detection of incremental changes, even if they occur slowly (Baena-Garca et al., 2006).

- **Adaptive Windowing (ADWIN)**

Albert Bifet and Ricard Gavald`a (2007) described a novel method for addressing distribution change and concept drift when acquiring knowledge from data sequences that may change over time. This algorithm, which compares the distributions of two detection windows, is called adaptive windowing (ADWIN). ADWIN effectively maintains a size window of recent items, ensuring that data distribution remains unchanged. By comparing the average of two sub-windows, this approach reveals drift. The window expands so long as there is no concept drift (Bifet A & Gavalda R, 2007).

- **Paired Learners**

Whereas DDM and EDDM concentrate on the error rate of an ML model, ADWIN compares two windows' distribution data to detect drift. The paired learners technique detects drift differently. This method is based on two learners: a stable learner and a reactive learner. The stable learner produces predictions based on all of its prior experience, whereas the reactive learner makes predictions based on its prior experience that occurred within a recent time window. Concept drift occurs when the model performance of the stable learner is inferior to that of the reactive learner for the same time window. When this happens, the stable learner must be updated (Bach & Maloof, 2008). The methods described above are often used to identify drift. The next stage is to identify a method for updating the model; this is also known as concept drift adaptation.

6.5 Concept drift adaptation

Concept drift adaptation refers to the procedures utilised to update a model once concept drift has occurred. The simplest, most straightforward response to concept drift is to retrain the model with the most recent data. This strategy requires a detection method to determine when the model should be updated. Additionally, a window that sorts the most recent data is required. The second drift adaptation strategy is to use an adaptive model, which can adjust itself based on the changing data, rendering retraining the model with the most recent data unnecessary. Such models are able to update themselves when data distributions change. Finally, using adaptive ensembles by using a set of learners is another concept drift adaptation strategy. Using specified voting rules, the outputs of each base classifier are combined to predict the newly arriving data. This method functions either by extending traditional ensemble methods or by establishing specific adaptive voting rules to address concept drift occurrence (Feng Gu, 2019).

6.6 Methodology

In the same way that medical devices require approval before being released to the public, AI models must also meet certain criteria in order to be considered safe for use. Performance metrics like sensitivity, specificity, and recall, for example, must be sufficiently high with low variability. Additionally, underlying biases for certain sub-populations must be identified and/or removed. However, once a piece of AI-based software has been approved, it must also be monitored to ensure that its predictions, diagnoses, and recommendations continue to operate safely for the population it was designed for. Hence, concept drift affecting the performance of ML models must be addressed by building a system to monitor ML performance and by updating such models when concept drift occurs. In this chapter, I investigate numerous batches of COVID-19-related primary care data released by the UK's Clinical Practice Research Datalink (CPRD). The CPRD is funded by the UK Department of Health’s observational and interventional research service. It utilises connected datasets and the UK’s health system to provide academics with access to anonymised, high-quality primary and secondary healthcare data (Clinical Practice Research, 2022). In addition, I investigate a dataset of SSc patients obtained from the Royal London Hospital (see chapter 3).

Here, it should be noted that drift can emerge in a variety of ways:

Performance drift: This refers to a change in the statistical performance of a model such as its accuracy, sensitivity, or recall.

Structural shift: This refers to a fundamental change in a model that is updated with new data on a regular basis, regardless of whether performance changes.

Class shift: This refers to a change in classes' underlying distribution.

Though all types of drift are of interest, the focus of this chapter is on performance drift, as this is regulators’ primary concern. Hence, I investigate a well-established drift detection approach that was first developed for streaming data but was modified for use with batch data here. I discuss the effects of drift detection on both performance metrics and model complexity. Additionally, I explore the effects of detection on abrupt drift and gradual drift.

6.6.1 Proposed Methods Protocol

Gama et al. (2004) created the DDM to assess model error rates and detect concept drift (see chapter 3). The DDM works based on the principle that a classifier's overall error rate remains constant unless concept drift occurs; in this instance, the error rate increases. To demonstrate this effect, I computed the classification error rate (p_t) and the standard deviation (s_t). Additionally, I held that p_{min} was the minimum error rate recorded for the classifier, and s_{min} was its standard deviation. Theoretically, p_{min} and s_{min} are two updated variables that work under the following conditions:

$$p_t + s_t < p_{min} + s_{min}$$

In the above condition, the model is operating normally with no alarms.

$$p_t + s_t \geq p_{min} + 2 * s_{min}$$

In the above condition, future drift is possible, and the model is now in the warning zone.

$$p_t + s_t \geq p_{min} + 3 * s_{min}$$

In the above condition, drift has occurred, and the model must be updated.

The error rate p_t was calculated as follows:

$$p_t = \frac{the\ number\ of\ patients\ \ predicted\ incorrectly}{the\ total\ number\ of\ patients} \quad (6.1)$$

$$s_t = \sqrt{\frac{p_t(1-p_t)}{n}} \quad (6.2)$$

where *n* is the total number of patients.

In this chapter, I adapted and implemented the DDM approach for use with batch data to monitor the performance of a random forest model. The error rate and standard deviation were computed for every incoming patient per batch. According to my proposed method, as long as the overall error rate decreases or remains constant, the model is not updated. When the overall error rate hits the drift level, the model is retrained by adding data from the warning batch to the drift batch to the new model (i.e., data from the warning level is added to the drift level). As explained in the following section, this approach was applied to the SSc dataset, simulated datasets, and the COVID-19 dataset. Because DDM focuses only on the error rate, I also investigated the effect of recall metric on drift detection.

This method, which explores the effect of recall metric on drift detection, is called DDM-OCI, which is a variant of DDM. This method focuses on minority class recall. As a result, instead of monitoring the error rate, as DDM does, DDM-OCI monitors recall. Essentially, this method detects drift when recall drops significantly. To implement this method, I computed the probability that correctly predicted patients had the true label (p_t) and the standard deviation (s_t) for every incoming patient per batch. DDM-OCI is nearly identical to standard DDM. The only difference is that DDM-OCI aims to maximise a model's recall instead of minimising error. Additionally, I held that p_{max} was the maximum p_t recorded of the classifier, and s_{max} was its standard deviation. Theoretically, p_{max} and s_{max} are two updated variables that work under the following conditions (Wang et al., 2013):

$$p_t - s_t \geq p_{max} - s_{max}$$

In the above condition, the model is operating normally with no alarms.

$$p_t - s_t < p_{max} - 2 * s_{max}$$

In the above condition, future drift is possible, and the model is in the warning zone.

$$p_t - s_t < p_{max} - 3 * s_{max}$$

In the above condition, there is a drift, and the model must be updated.

The error rate p_t is the probability that correctly predicted patients have the correct label:

$$p_t = \frac{the\ number\ of correctly\ predicted\ \ patients\ having\ true\ label}{the\ total\ number\ of\ minority{-}class\ label\ patients} \qquad (6.3)$$

$$s_t = \sqrt{\frac{p_t(1-p_t)}{n}} \quad (6.4)$$

$n: the\ total\ number\ of\ minority - class\ label\ patients$

As with DDM, I implemented DDM-OCI for usage with batch data to monitor the performance of a random forest model. The probability that correctly predicted patients had the true label and standard deviation was computed for every incoming patient per batch. According to my proposed method, as long as recall increases or stays constant, the model is not updated. When the recall drops the drift level, the model is retrained by adding data from the warning batch to the drift batch to the new model (i.e., data from the warning level is added to the drift level).

In this study, due to the imbalanced data wherein the classes were not represented equally, I implemented under-sampling to ensure the balance of both classes in order to train the random forest model. This method works by randomly selecting a number of patients from the majority class that is equal to the number of patients from the minority class in order to balance the training dataset. The under-sampling method is illustrated in figure 6.3, which displays how the training dataset was balanced after the under-sampling procedure to avoid bias towards the majority class.

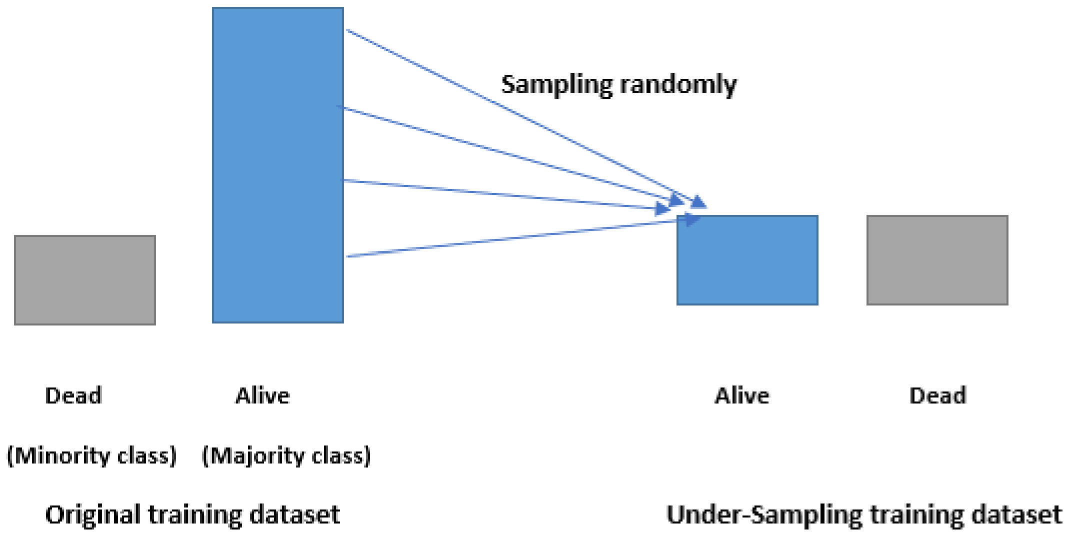
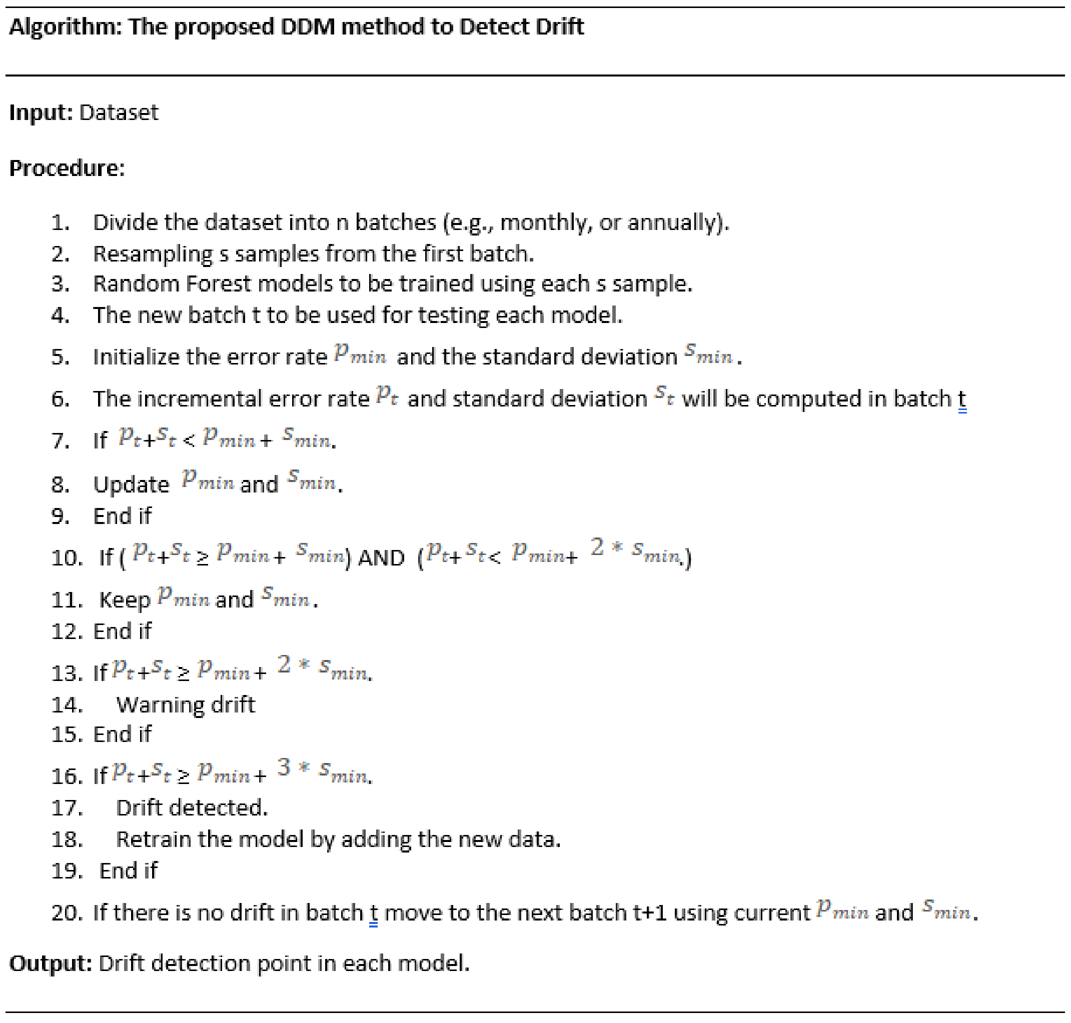


Figure 6.3: *Under-Sampling Method to Balance Imbalanced Training Dataset*

In summary, the above approaches, DDM and DDM-OCI, were implemented in this study to monitor the performance of the random forest model (see chapter 3) in classifying COVID-19 and SSc patients after under-sampling was used to balance the training dataset. In addition, these methods were applied to two simulated datasets, one with abrupt drift and the other with incremental drift, as explained in the following section. DDM and DDM-OCI are crucial tools for maintaining ML model performance over time.

Figures 6.4 and 6.5 provide a summary of the approaches proposed in this study.



Algorithm: The proposed DDM method to Detect Drift

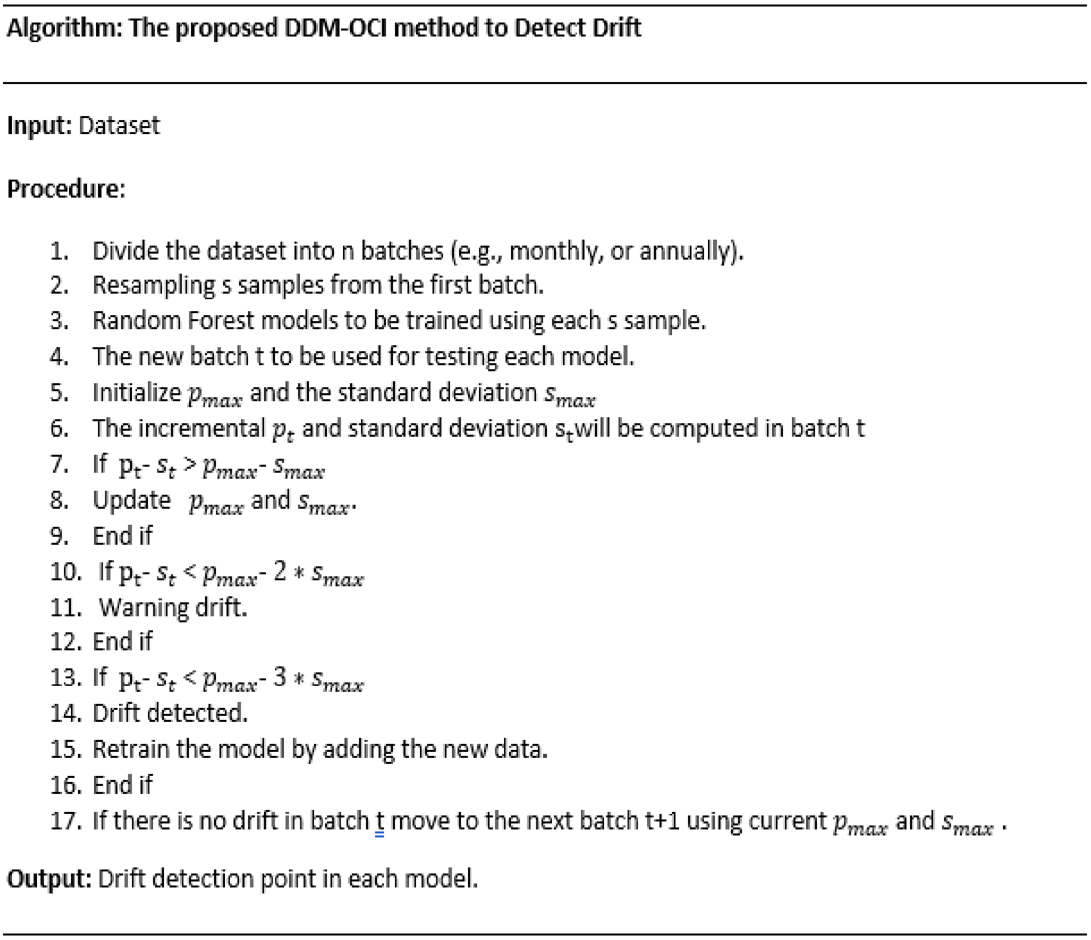
Input: Dataset

Procedure:

```
1. Divide the dataset into n batches (e.g., monthly, or annually).
2. Resampling s samples from the first batch.
3. Random Forest models to be trained using each s sample.
4. The new batch t to be used for testing each model.
5. Initialize the error rate $p_{min}$ and the standard deviation $s_{min}$.
6. The incremental error rate $p_t$ and standard deviation $s_t$ will be computed in batch t
7. If $p_t$+$s_t$ < $p_{min}$+ $s_{min}$.
8. Update $p_{min}$ and $s_{min}$.
9. End if
10. If ( $p_t$+$s_t$ ≥ $p_{min}$+ $s_{min}$) AND ($p_t$+ $s_t$< $p_{min}$+ $2 * s_{min}$.)
11. Keep $p_{min}$ and $s_{min}$.
12. End if
13. If $p_t$+$s_t$ ≥ $p_{min}$+ $2 * s_{min}$.
14.     Warning drift
15. End if
16. If $p_t$+$s_t$ ≥ $p_{min}$+ $3 * s_{min}$.
17.     Drift detected.
18.     Retrain the model by adding the new data.
19. End if
20. If there is no drift in batch t move to the next batch t+1 using current $p_{min}$ and $s_{min}$.
```

Output: Drift detection point in each model.

Figure 6.4: *DDM Algorithm*



Algorithm: The proposed DDM-OCI method to Detect Drift

Input: Dataset

Procedure:

```
1. Divide the dataset into n batches (e.g., monthly, or annually).
2. Resampling s samples from the first batch.
3. Random Forest models to be trained using each s sample.
4. The new batch t to be used for testing each model.
5. Initialize $p_{max}$ and the standard deviation $s_{max}$
6. The incremental $p_t$ and standard deviation $s_t$will be computed in batch t
7. If $p_t - s_t > p_{max} - s_{max}$
8. Update $p_{max}$ and $s_{max}$.
9. End if
10. If $p_t - s_t < p_{max} - 2 * s_{max}$
11. Warning drift.
12. End if
13. If $p_t - s_t < p_{max} - 3 * s_{max}$
14. Drift detected.
15. Retrain the model by adding the new data.
16. End if
17. If there is no drift in batch t move to the next batch t+1 using current $p_{max}$ and $s_{max}$ .
```

Output: Drift detection point in each model.

Figure 6.5: *DDM-OCI Algorithm*

6.6.2 Datasets

I explored three datasets in this chapter: a simulated dataset, a COVID-19 dataset, and an SSc dataset.

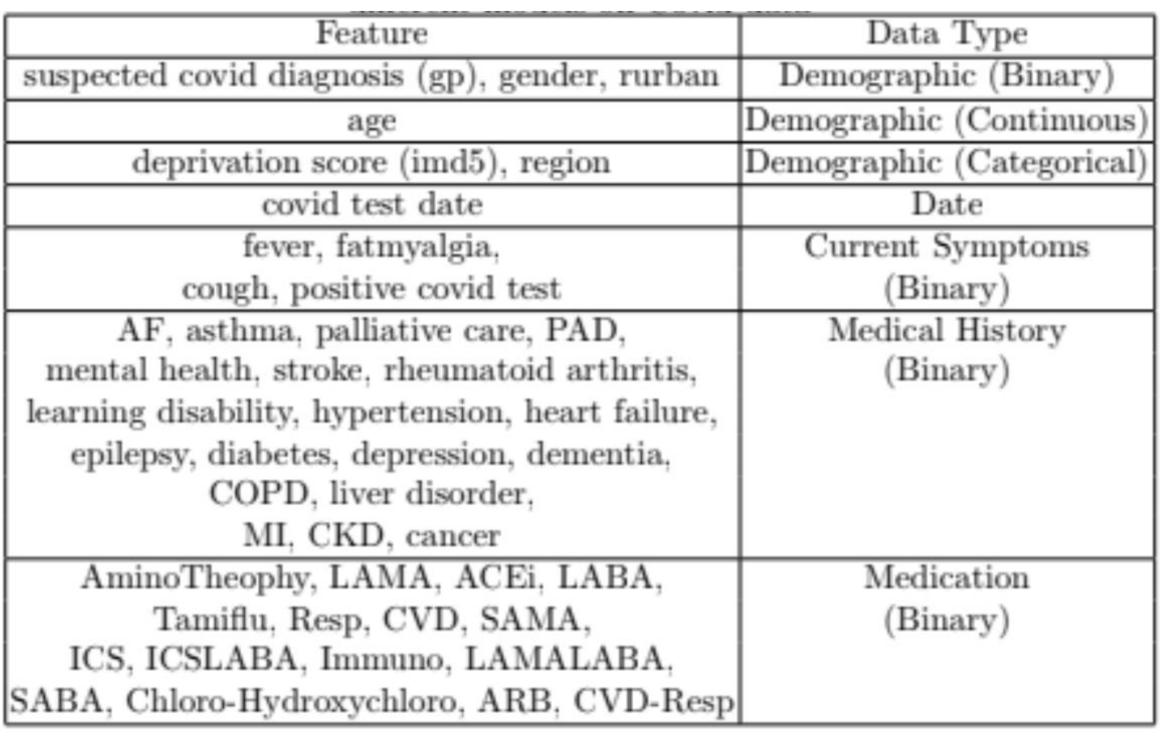
- **Simulated Dataset**

Using Agrawal's data generator, I created two datasets (Agrawal et al., 1993), which contained six numerical features, three categorical features and one binary class. In addition, each dataset contained 60,000 instances. In order to validate my suggested approach, I created sudden drift for the first dataset and incremental drift for the second dataset. In addition, each dataset was divided into 12 batches. In the following sections, the experiments I conducted with these datasets and my findings are described.

- **Synthetic COVID-19 Dataset**

This synthetic COVID-19 dataset was derived completely from authentic, anonymised primary care patient data retrieved from the CPRD Aurum database, which was provided by the UK's CPRD. This synthetic dataset contained information on patients who sought primary care with symptoms that could potentially be related to COVID-19. This data contained information on both social and clinical risk factors. The following table uses symptoms, drugs, and demographic information as features in this dataset, which includes 779,546 patients. Additionally, this dataset, which contained information collected from December 2019 to April 2021, was divided into monthly batches.

Table 6.1: *Synthetic COVID-19 Dataset Features and Data Type*



Feature	Data Type
suspected covid diagnosis (gp), gender, rurban	Demographic (Binary)
age	Demographic (Continuous)
deprivation score (imd5), region	Demographic (Categorical)
covid test date	Date
fever, fatmyalgia, cough, positive covid test	Current Symptoms (Binary)
AF, asthma, palliative care, PAD, mental health, stroke, rheumatoid arthritis, learning disability, hypertension, heart failure, epilepsy, diabetes, depression, dementia, COPD, liver disorder, MI, CKD, cancer	Medical History (Binary)
AminoTheophy, LAMA, ACEi, LABA, Tamiflu, Resp, CVD, SAMA, ICS, ICSLABA, Immuno, LAMALABA, SABA, Chloro-Hydroxychloro, ARB, CVD-Resp	Medication (Binary)

- **SSc Dataset**

This dataset, which was provided by the Royal London Hospital (see chapter 3), was used to detect concept drift. This dataset contained information on 677 patients and was divided into 6 batches. The experiments performed on the aforementioned datasets using the aforementioned proposed methods are explained in the following section.

6.6.3 Experiments

This section describes the experiments conducted in order to detect drift in the random forest model, which may become outdated when new data becomes available over time. Initially, as mentioned previously, I investigated the implications of drift detection using two distinct performance measures: error rate, using the standard DDM rate, and recall, using the DDM-OCI. In addition, I investigated whether drift is abrupt and occurs within a short time (e.g. changes associated with using a new clinical metric) or whether it is gradual and occurs slowly over time (e.g. changes associated with an ageing population). Furthermore, I first implemented a simple logistic classifier, then a NB model, and ultimately a random forest model, the complexity of which increased as the number of parameters increased. I implemented a number of models because complex models are likely more susceptible to overfitting, resulting in the discovery of more erroneous drift points.

- **Simulated Datasets**

To classify the class for both simulated datasets, I applied the random forest model (see chapter 3) to the first 5,000 instances (first batch). As previously mentioned, these datasets were divided into batches. In accordance with my experiment, I tested the outcome model on the second batch and employed the proposed drift detection approaches to detect drift. The model was not updated if there was no drift, and it was tested on the following batch. Nonetheless, a new model was taught using fresh data if drift was detected. The process of drift detection method assessment is depicted in Figure 6.4.

- **SSc Dataset**

Using this dataset, I implemented the random forest model to classify death (to classify patients who died or not) and PAH (to classify patients who developed PAH or not). In accordance with my experiment, I implemented the first random forest model using the first batch (patients’ data from 1995 to 1998). I tested the outcome model on the second batch and employed the proposed drift detection approaches to detect drift. The model was not updated if there was no drift, and it was tested on the following batch. The process of drift detection method assessment is depicted in Figure 6.4.

- **Synthetic COVID-19 Dataset**

Using this dataset, I implemented the random forest model to classify the death feature and determine whether patients were dead or alive. Furthermore, because the majority class ‘death feature’ was linked to living patients, an under-sampling method, as described above, was implemented to prevent bias towards living patients. When the random forest model was implemented, the training dataset was balanced so that the number of living patients was equal to the number of dead patients. As previously mentioned, this dataset was divided into monthly batches. In accordance with my experiment, I built the first random forest model to the patients from December 2019 until March 2020. I tested the outcome model on the April 2020 batch and employed the proposed drift detection approaches. The model was not updated if there was no drift, and it was tested on the following batch. However, a new model was taught using the fresh data if drift was detected. The process of drift detection method assessment is depicted in Figure 6.6.

The main protocol I followed to carry out my experiments is illustrated in Figure 6.6.

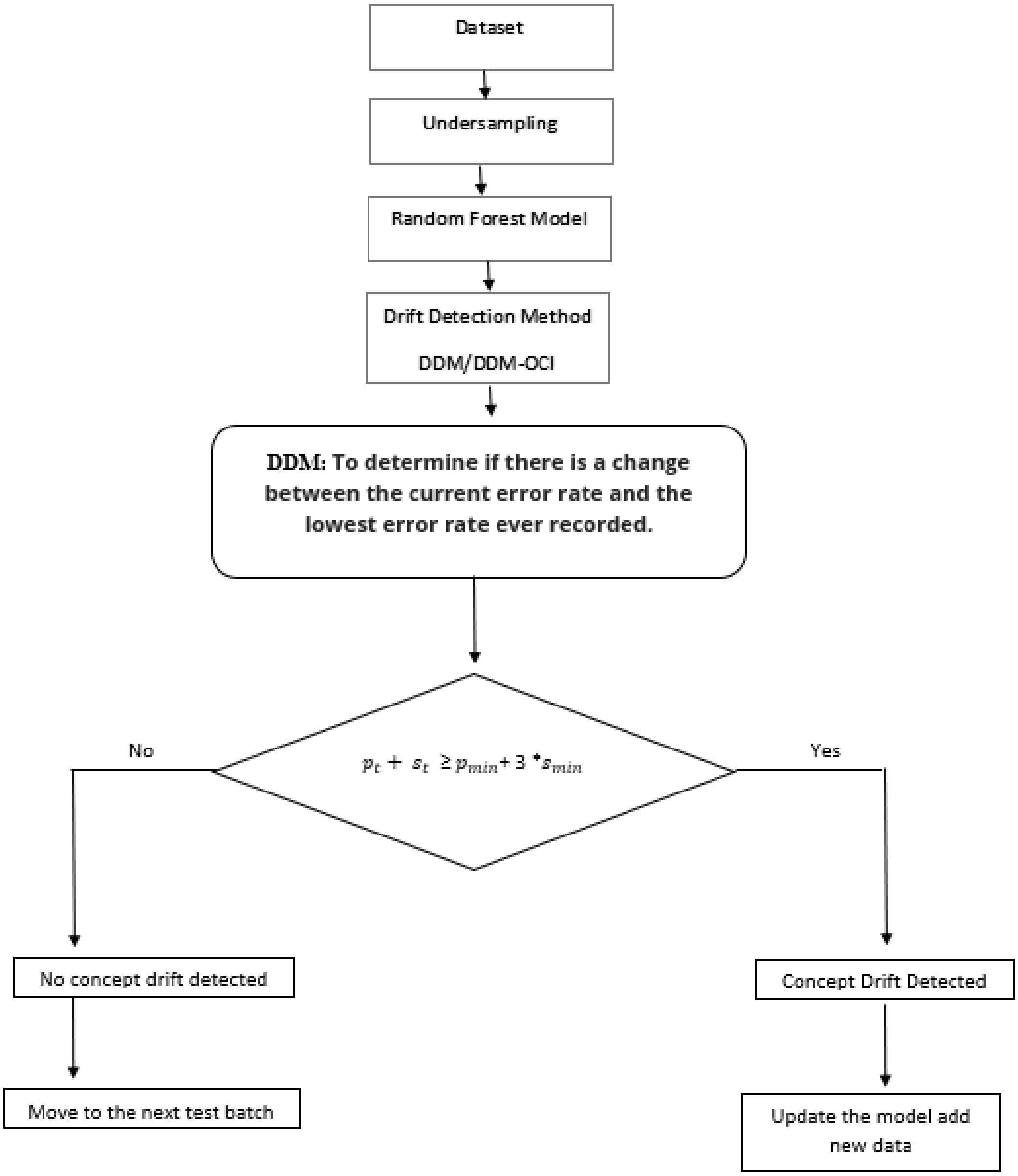

Dataset
Undersampling
Random Forest Model
Drift Detection Method
DDM/DDM-OCI
DDM: To determine if there is a change between the current error rate and the lowest error rate ever recorded.
$p_t + s_t \geq p_{min} + 3 * s_{min}$
No
Yes
No concept drift detected
Move to the next test batch
Concept Drift Detected
Update the model add new data

Figure 6.6: *Main Protocol Used in My Experiments*

Figure 6.7 displays the implementation of the DDM-OCI approach.

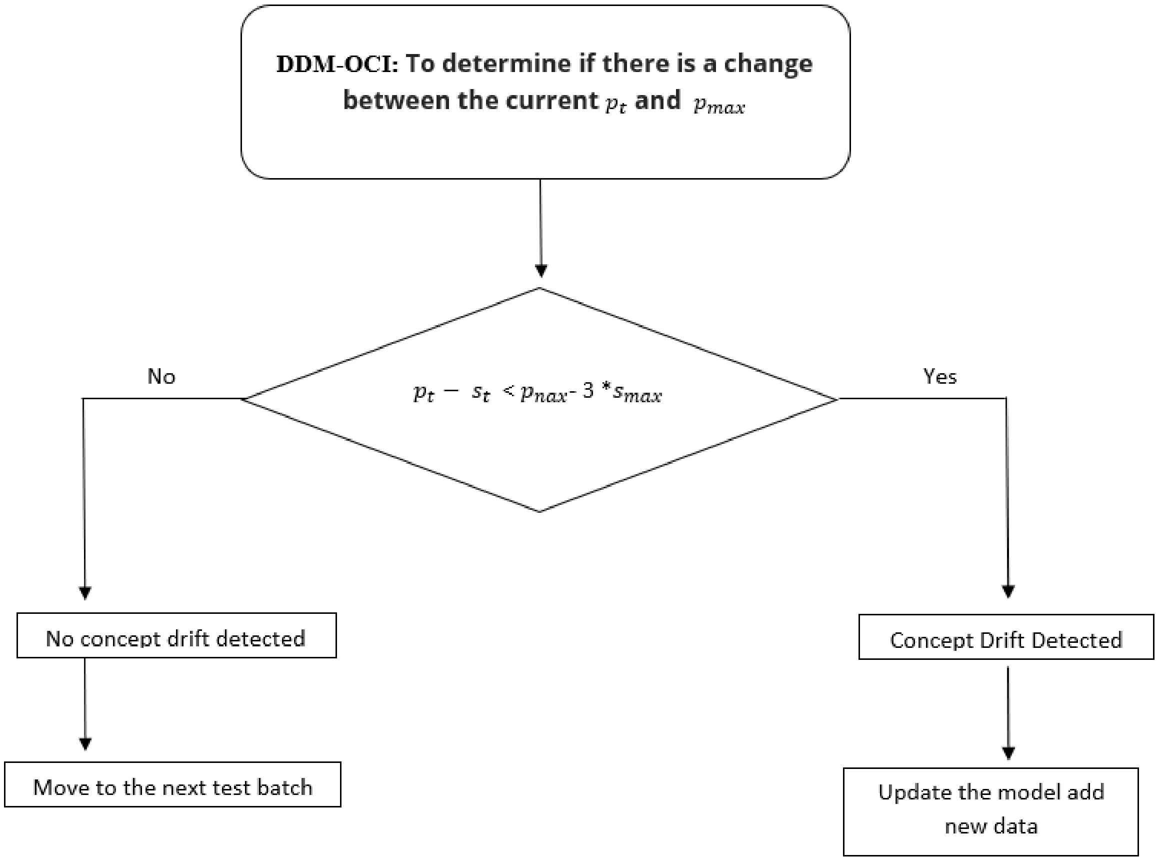

DDM-OCI: To determine if there is a change between the current p_t and p_{max}
No
$p_t - s_t < p_{nax}$- 3 *s_{max}
Yes
No concept drift detected
Move to the next test batch
Concept Drift Detected
Update the model add new data

Figure 6.7: *DDM-OCI Protocol*

Finally, figure 6.8 illustrates the key methodology I utilised to assess whether an abrupt or gradual drift strategy should be used.

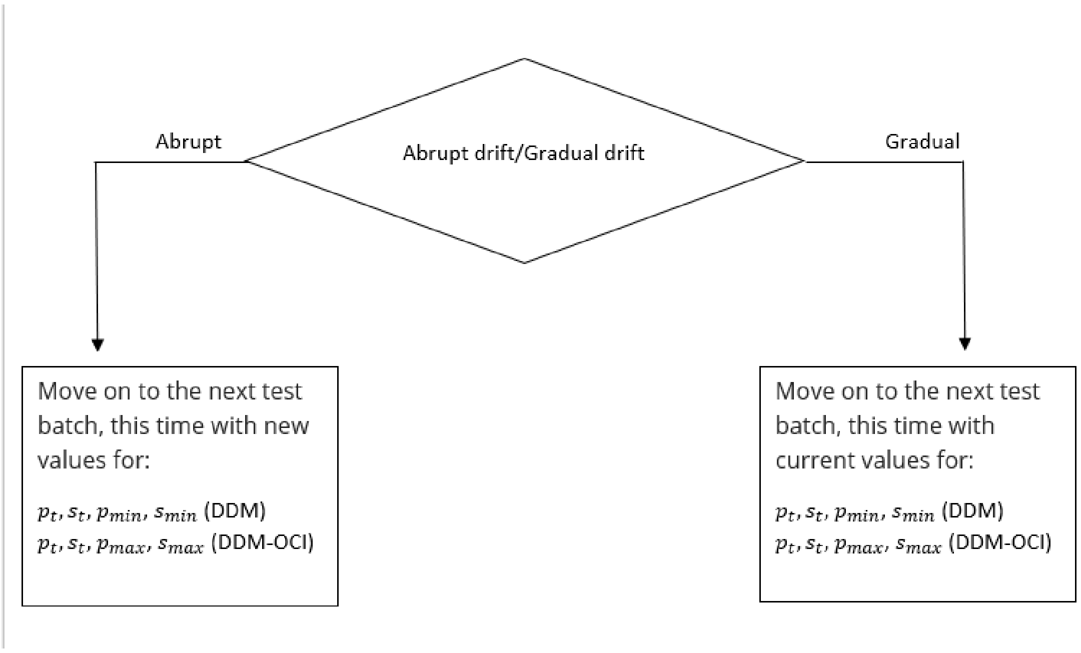

Abrupt
Abrupt drift/Gradual drift
Gradual
Move on to the next test batch, this time with new values for:
$p_t, s_t, p_{min}, s_{min}$ (DDM)
$p_t, s_t, p_{max}, s_{max}$ (DDM-OCI)
Move on to the next test batch, this time with current values for:
$p_t, s_t, p_{min}, s_{min}$ (DDM)
$p_t, s_t, p_{max}, s_{max}$ (DDM-OCI)

Figure 6.8: *Types of Drift Implemented*

Following the discussion of the experiments I conducted in this study, the section below presents the results of these methods when applied to my datasets.

6.7 Results

- **Simulated Data Results**

As previously stated, one simulated dataset simulated abrupt drift, and the other dataset simulated incremental drift. For these simulated datasets, the performance of a random forest model based on the error rate is depicted in the following figures. New batches of data were introduced over time to test performance, and this shows the effect an outdated model has on the error rate. The figure on the left (Figure 6.9a) clearly illustrates that there was a significant change after batch 5, when the error rate increased dramatically. In addition, the figure on the right (Figure 6.9b) clearly indicates that the error rate gradually increased after batch 3.

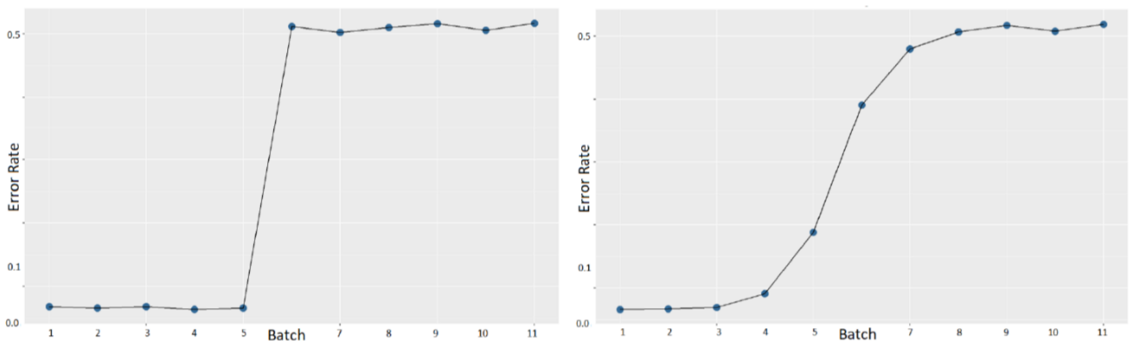


Figure 6.9: *a) Performance Drift (Error Rate) for a Simulated Abrupt Change b) Performance Drift (Error Rate) for a Simulated Incremental Change for Incoming Batches of Data Created Over Time*

The proposed approaches to drift detection were applied to these two datasets in accordance with the previously described experimental protocol.

The following figures (Figure 6.10 and Figure 6.11) illustrate the performance of an updated random forest model based on the DDM method, which utilises error rate to detect drift, and the DDM-OCI, which utilises recall to detect drift. According to my findings, the batch observed drift is depicted as vertical lines.

Figure 6.10a shows the drift detected by the DDM method on the gradual simulated dataset. The random forest model was created using the initial 5,000 cases and tested on subsequent batches. The DDM detected the first drift in batch 5, as shown in figure 6.10a. As a result, the random forest model was updated using the fresh batch of data and tested on subsequent batches. Additionally, the DDM detected the second instance of drift in batch 6, as shown in figure 6.10a. As a result, the random forest model was updated using the fresh batch of data and tested on subsequent batches. Figure 6.10a clearly shows a sudden improvement in error rates following this update. The last instance of drift was detected in batch 11, as shown in figure 6.10a.

Figure 6.10b shows the drift that was detected by the DDM-OCI method on the gradual simulated dataset. The random forest model was created using the initial 5,000 cases and tested on the subsequent batches. Like DDM, DDM-OCI found drift in batches 5, 6, and 11. However, as illustrated in figure 6.10b, the DDM-OCI discovered further drift in batch 8.

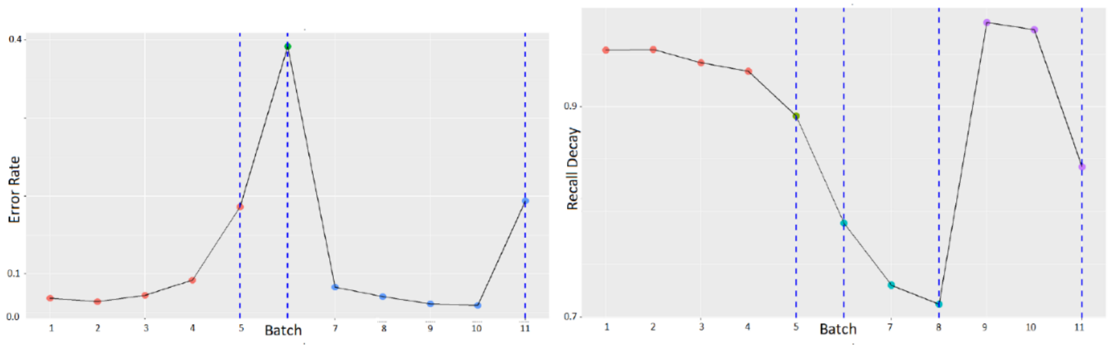


Figure 6.10: *a) DDM Method on Incremental Drift b) DDM-OCI Method on Incremental Drift*

Figure 6.11a demonstrates the drift detected by the DDM method on the abrupt simulated dataset. Like the incremental simulated dataset, the random forest model was created using the initial 5,000 cases and tested on subsequent batches. Figure 6.11a shows that DDM found only one instance of drift, in batch 5. As a result, the random forest model was updated with a new batch of data and tested on subsequent batches. As depicted in figure 6.9a, the DDM was unable to detect any more drift, and the error rate suddenly improved. Figure 6.11b shows the drift that was detected by the DDM-OCI method on the abrupt simulated dataset. The DDM-OCI detected the same and only drift found by the DDM in batch 5. As before, recall suddenly improved.

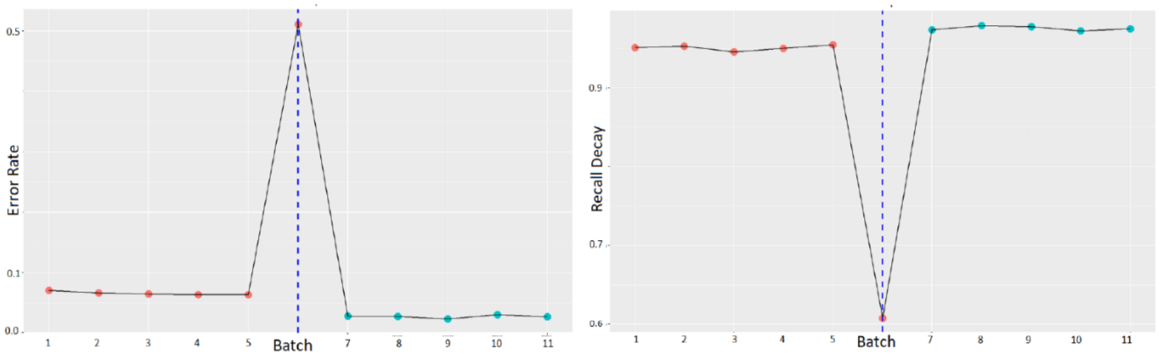


Figure 6.11*: a) DDM Method on Abrupt Drift b) DDM-OCI Method on Abrupt Drift*

For these simulated datasets, it appears that the choice of metric was irrelevant, as drift was appropriately identified using both recall and error. This may be due to the fact that the simulated data was balanced. The following section discusses the outcomes of my experiments with significantly imbalanced COVID-19 data.

- **SSc Data Results**

The proposed approaches to drift detection were also applied to the SSc data in accordance with the previously described experimental protocol. Figures 6.12 and 6.13 show the results of the DDM method applied to the SSc dataset. The random forest model was developed utilising patient data from 1995 to 1998 to classify death (i.e., patients who died or not) and PAH (i.e., patients who developed PAH or not). The model was tested on subsequent annual batches.

Figures 6.12 and 6.13 illustrate the outcomes of the concept drift approach applied to this dataset in accordance with my experimental procedure. Figure 6.12 demonstrates a drift in 2002 due to a significant increase in the model error rate; therefore, the model was updated accordingly. The vertical line represents the drift in 2002. Figure 6.13 demonstrates that this dataset was consistent across all batches when PAH classification was performed.

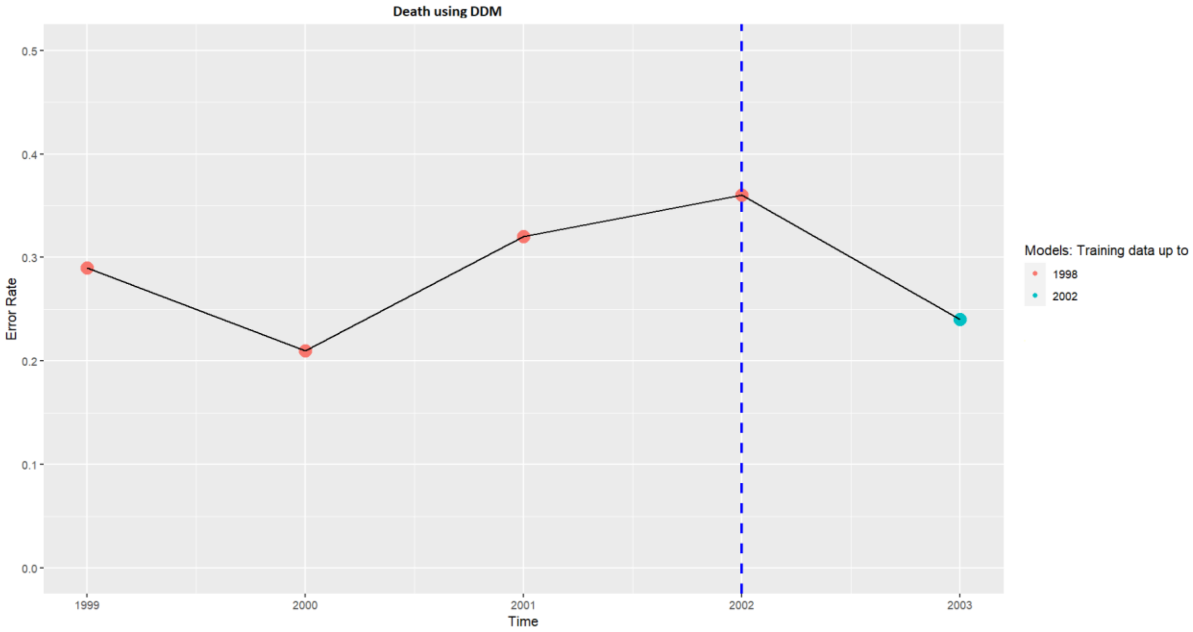


***Figure 6.12**: DDM Random Forest Model on SSc Data to Classify Death*

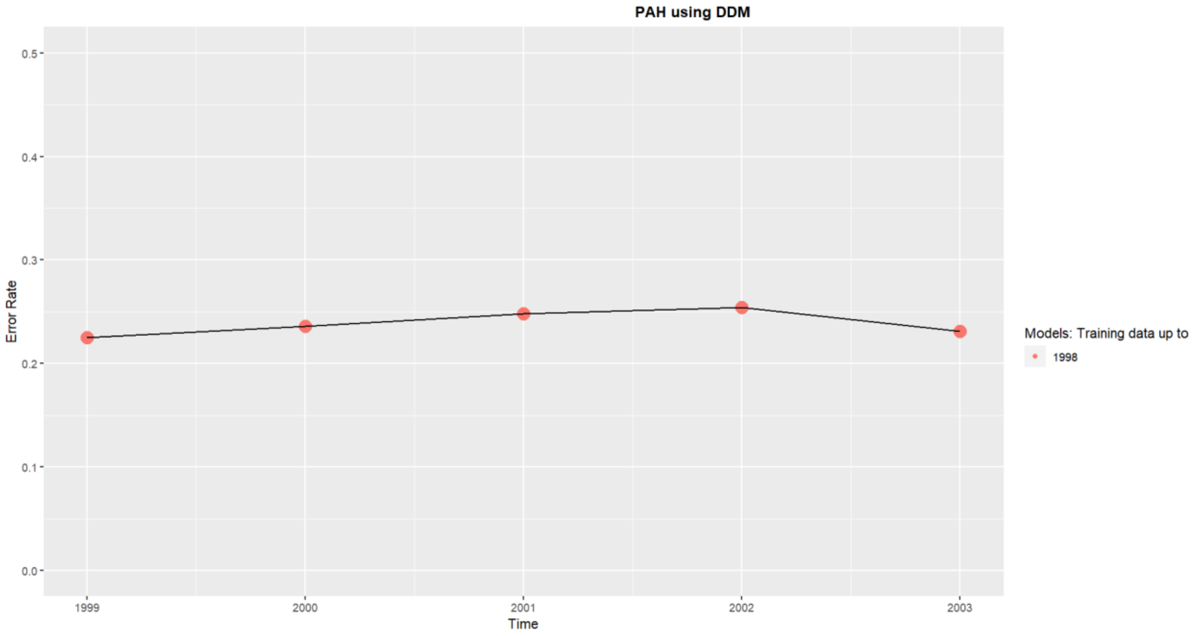


Figure 6.13: *DDM Random Forest Model on SSc Data to Classify PAH*

Also, I have used DDM-OCI to classify death. Figure 6.14 demonstrates that there is a drift in 2002 due to a significant decrease in the model recall.

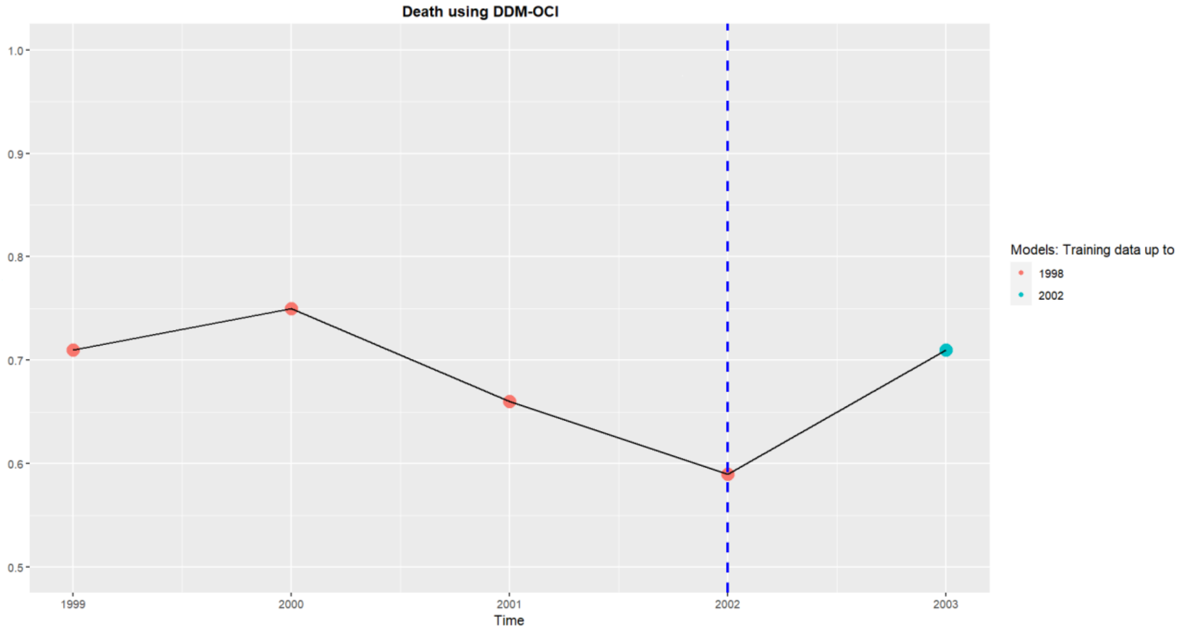


Figure 6.14: *DDM-OCI Random Forest Model on SSc Data to Classify Death*

I was unable to reimplement my models in chapters 4 and 5 to identify a drift due to the limited size of my dataset, and the models will need to be updated to handle concept drift so I can adjust my models in the future to account for it. Nonetheless, I ran consensus clustering and latent class analysis on the data before and after the drift to determine whether any features had changed that might cause the drift. The following tables (Table 6.3 and Table 6.4) show the results of the consensus clustering method and latent class analysis for patients' data before and after the drift. It was observed that the percentage of skin thickness increased in all groups after the drift compared to groups before the drift for both approaches. In the consensus cluster method, the mean value of Cr in group 3 for patients after the drift was lowest when compared to the other groups and the groups for patients before drift, as shown in Table 6.3. In addition, the mean value of DLCO in group 3 for patients after the drift was highest when compared to the other groups and the groups for patients before the drift. When the mean value of Cr in group 3 for patients after the drift was the lowest, the mean value of FVC in group 3 for patients after the drift was highest compared to other groups and the groups for patients before the drift. For the latent class analysis approach, the mean value of Hb was highest in group 3 for patients after the drift, while the mean value of Hb was lowest in group 3 for patients before the drift. As demonstrated in Table 6.4, when the mean of Cr was lowest in group 3 for patients after the drift, the mean value of FVC was highest in comparison to other groups and groups for patients before the drift. According to the results, these modifications might be the cause of identifying a drift in 2002, as the characteristics of patients before and after 2002 differ. In a nutshell, there are certain differences in the personality characteristics of the patients features in the groups before and after 2002 that might cause AI models to drift. These variations might be the result of a change in how the hospital collects information or anything else.

Table 6.3 Consensus Clustering Method for patients' data before the drift and patients' data after the drift.

Consensus Clustering						
	Before Drift (1995-2002)			After Drift (2002-2003)		
	Proportions			Proportions		
	Group1	Group2	Group3	Group1	Group2	Group3
No Skin Thickening	0.61	0.60	0.72	0.41	0.36	0.51
Skin Thickening	0.39	0.40	0.28	0.59	0.64	0.49
Male	0.14	0.11	0.14	0.24	0.22	0.07
Female	0.86	0.89	0.86	0.76	0.78	0.93
	Means			Means		
	Group1	Group2	Group3	Group1	Group2	Group3
Hb	12.67	12.74	12.61	12.90	12.57	13.16
Cr	90.24	91.36	87.53	90.49	89.8	73.25
FVC	87.24	90.33	88.58	88.52	89.02	97.25
DLCO	64.87	65.80	63.43	65.13	69.62	71.50
Age	47.90	48.68	48.17	49.23	44.20	49.33

Table 6.4 Latent Class Analysis Method for patients' data before the drift and patients' data after the drift.

Latent Class Analysis						
	Before Drift (1995-2002)			After Drift (2002-2003)		
	Proportions			Proportions		
	Group1	Group2	Group3	Group1	Group2	Group3
No Skin Thickening	0.75	0.84	0.10	0.39	0.06	0.65
Skin Thickening	0.25	0.16	0.90	0.61	0.94	0.35
Male	0.17	0.07	0.12	0.27	0.17	0.13
Female	0.83	0.93	0.88	0.73	0.83	0.87
	Means			Means		
	Group1	Group2	Group3	Group1	Group2	Group3
Hb	12.71	12.93	10.86	12.86	12.49	13.23
Cr	85.67	78.00	196.25	81.79	96.95	76.46
FVC	82.71	99.32	93.63	72.65	99.96	107.11
DLCO	61.35	69.95	72.74	51.79	82.00	75.70
Age	46.93	51.27	47.48	46.67	45.45	52.95

- **COVID-19 Data Results**

The proposed approaches to drift detection were applied to this data in accordance with the previously described experimental protocol. Figures 6.15 and 6.16 illustrate how the choice of performance metric affects drift detection in COVID-19 data. The random forest model was developed utilising patient data collected between December 19 and March 21 in order to predict risk of death based on symptoms, medications, and medical history. The model was then tested on subsequent batches.

Figure 6.15 illustrates the drift detected by the DDM method on the COVID-19 dataset. The DDM detected drift in April 2020, as shown in figure 6.15. As a result, the random forest model was updated using the fresh batch of data and tested on subsequent batches. Additionally, the DDM detected a second instance of drift in May 2020, as shown in figure 6.15. As a result, the random forest model was again updated using a fresh batch of data and tested on subsequent batches. The outcome model remained steady until March 2021, when the DDM detected drift, and the model was updated with new data. The DDM immediately identified another instance of drift in April 2021.

Figure 6.16 demonstrates the drift that was detected by the DDM-OCI method on the COVID-19 dataset. The DDM-OCI detected drift in August 2020, as shown in figure 6.16. As a result, the random forest model was updated using a fresh batch of data and tested on subsequent batches. Additionally, the DDM-OCI detected a second instance of drift in October 2020, as shown in figure 6.16. Likewise, the DDM-OCI detected drifts in January and February of 2021.

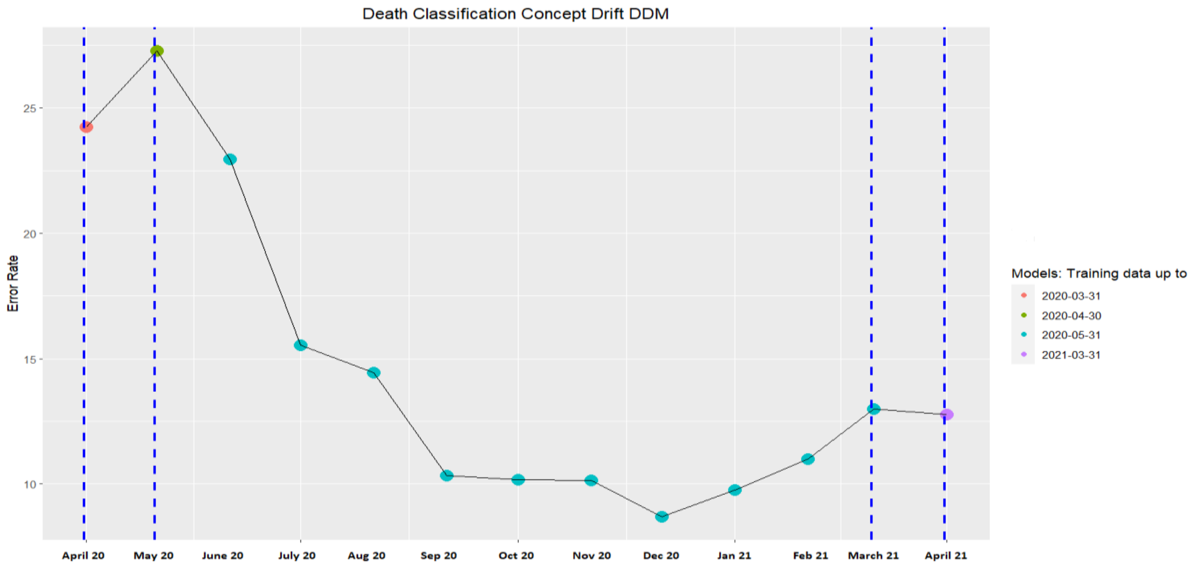


Figure 6.15: *DDM on COVID-19 Data*

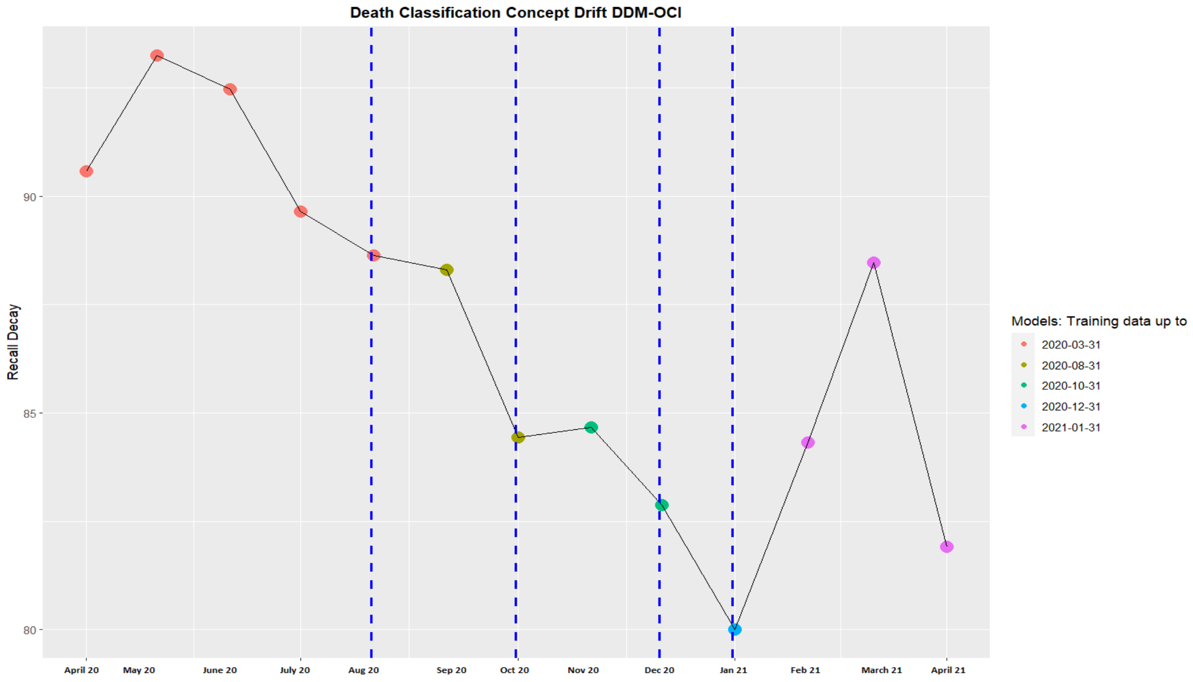


Figure 6.16: *DDM-OCI on COVID-19 Data*

In contrast to the simulated data, the metric here significantly affected where drift was discovered. The COVID-19 data was extremely unbalanced, which may have affected this. Before selecting metrics to detect performance drift, determining the appropriate categories of change is crucial. In some situations, a change in false positives (leading to unnecessary interventions) may be more relevant than a change in false negatives, whilst in other situations (e.g. screening populations for follow-up tests), the reverse may be true.

Although the focus of my investigation was on incremental drift, I did implement the aforementioned protocol for abrupt change on the COVID-19 data. Figure 6.17 illustrates the outcome of my experiment, in which this methodology identified three further drifts in September 2020, Dec 2020, and January 2021. It is crucial to note that my methods were able to detect some of the anticipated drifts, such as those that took place between April and May 2020, when the first nationwide lockdown occurred; in January 2021, when the second nationwide lockdown occurred; and in April 2021, the date by which 50% of people over 65 years old had received both vaccine doses.

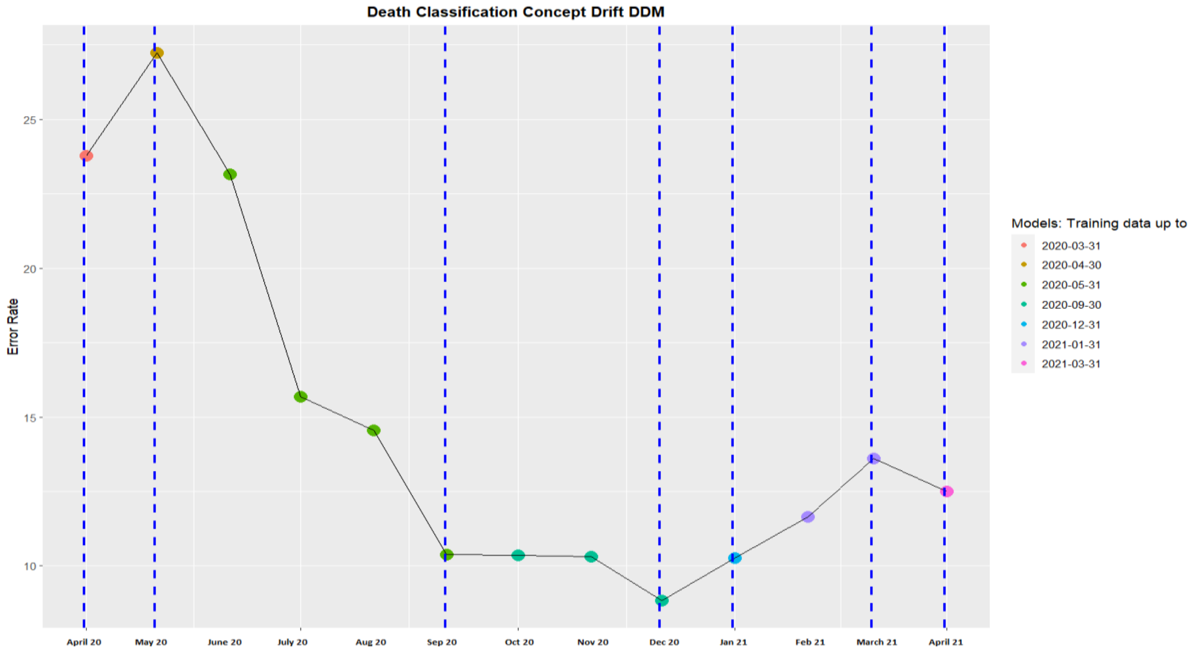


Figure 6.17: Abrupt DDM on COVID-19 Data

I also opted to investigate the effect of model selection on drift detection, focusing on whether models with a greater number of parameters are more susceptible to overfitting and, consequently, to detecting drift. Figure 6.18, figure 6.19, and figure 6.20 represent the outcomes of the three distinct models (LR, NB, and random forest) applied to the COVID-19 data. It has been noted that many of the drift points, particularly in the LR and random forest models, have considerable agreement. Additionally, I discovered that alternative models have less of an effect on drift detection than metric selection. Poorly fitted models, however, can lead to erroneous drift detection, which leads to less model stability due to unnecessary updating.

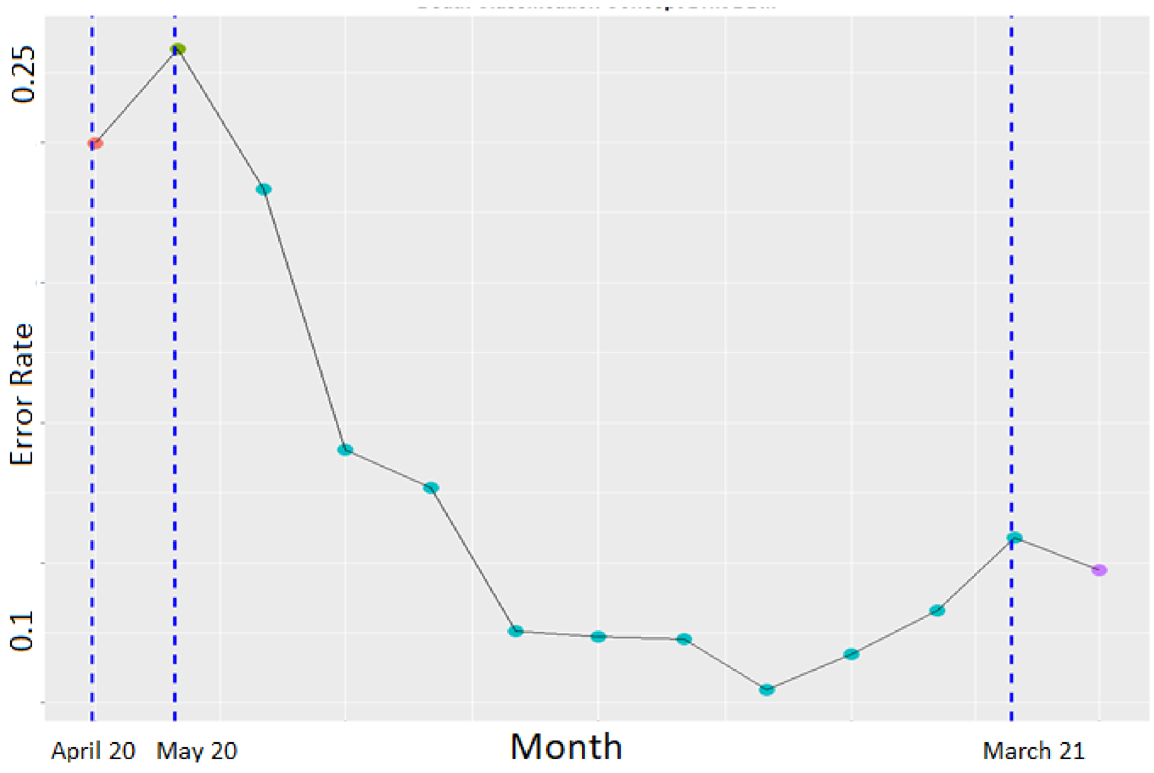


Figure 6.18: DDM Logistic Regression on COVID-19 Data

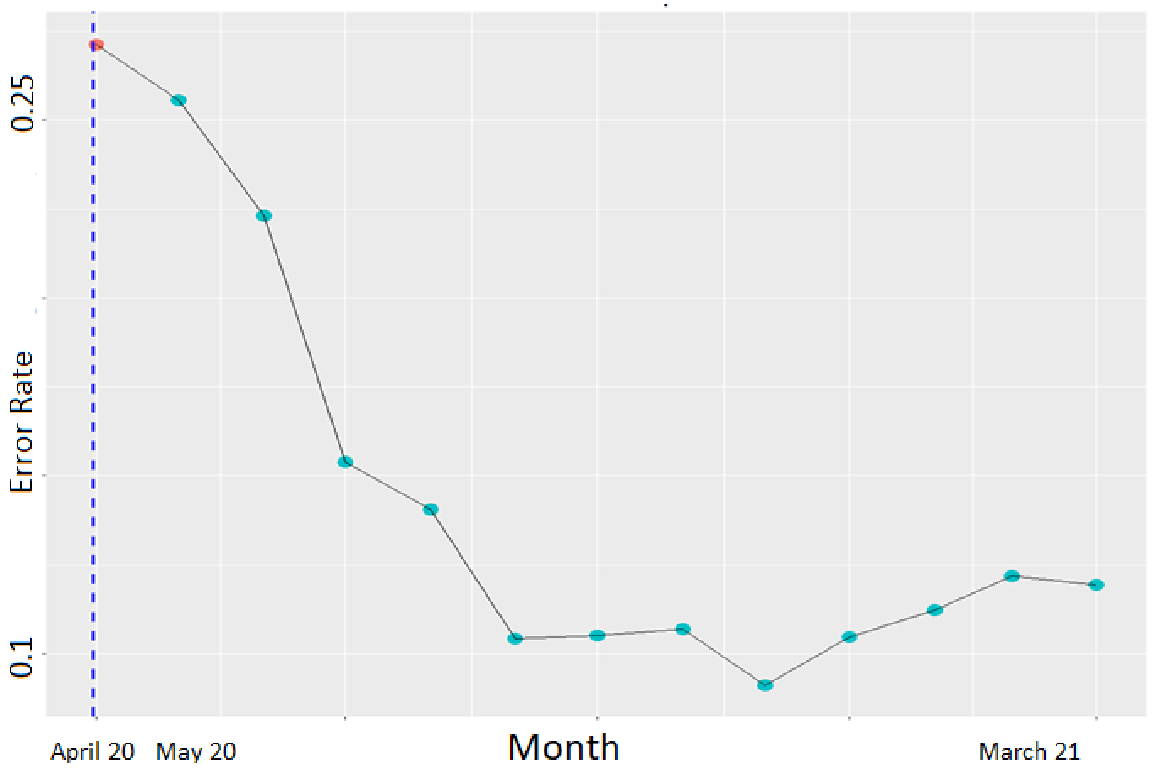


Figure 6.19: DDM Naive Bayes on COVID-19 Data

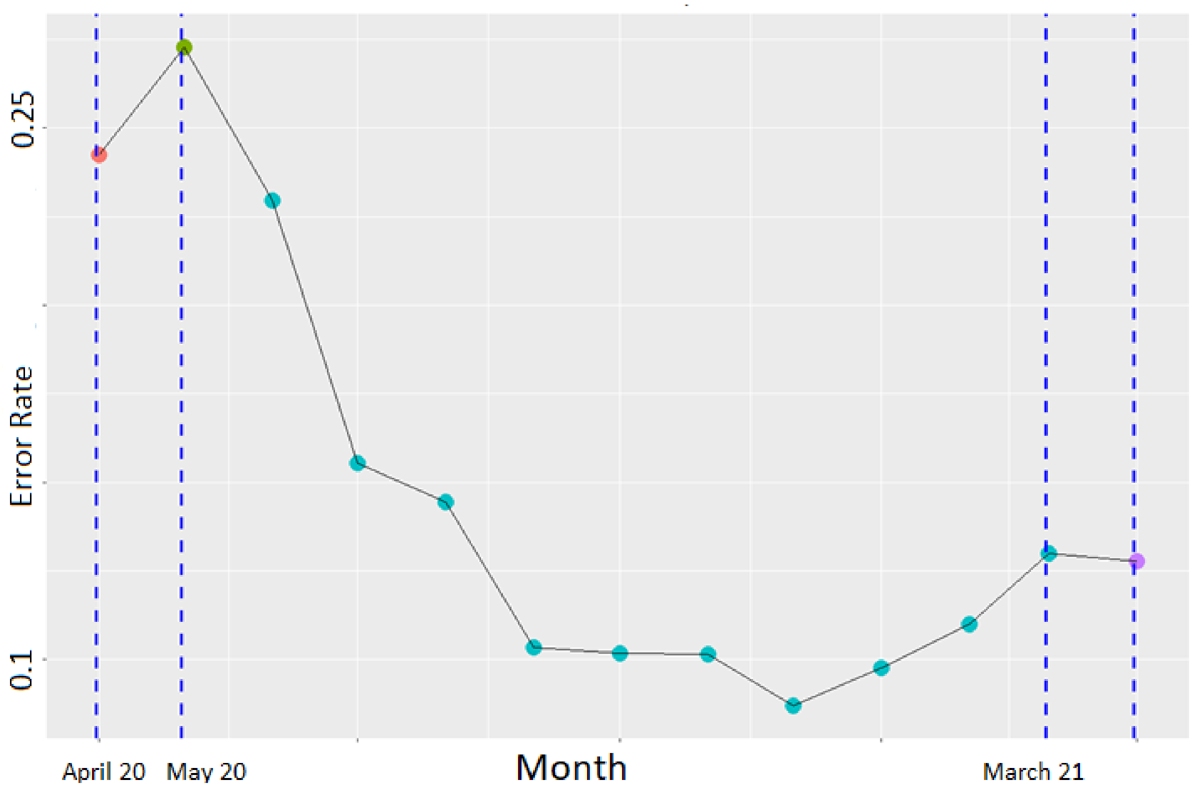
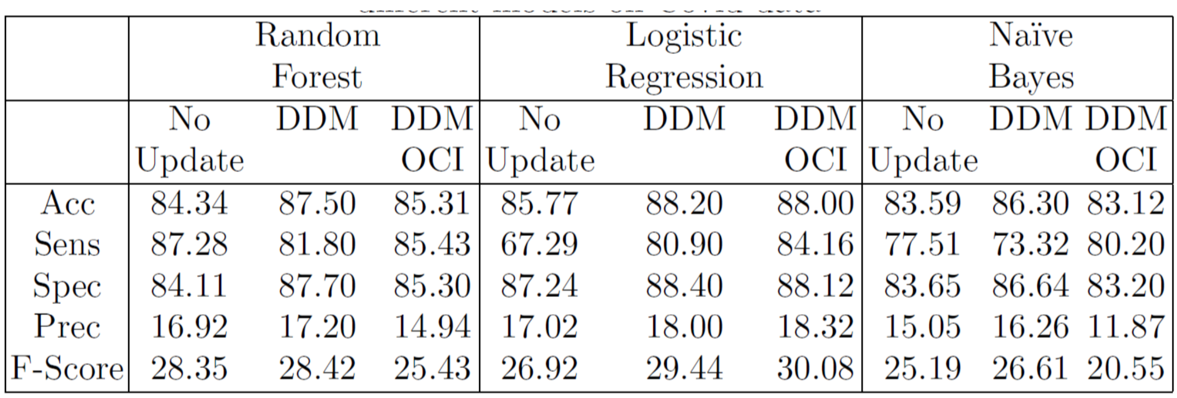


Figure 6.20: *DDM Random Forest on COVID-19 Data*

In addition, I evaluated the overall usefulness of concept drift detection by comparing numerous performance measures across the final set of approaches where drift was or was not detected, as shown in Table 6.2. It has been noted that the usage of drift detection and updating has increased performance metrics for numerous models and metrics. However, this is not always the case. For example, the DDM and DDM-OCI both work to decrease sensitivity when applied to the random forest model. Additionally, it is abundantly evident that the DDM surpasses the DDM-OCI in terms of accuracy as the DDM uses error rate to detect drift. By contrast, the DDM-OCI performs better in terms of sensitivity across all models as it uses the recall metric. Finally, these findings show that, while detecting drift and updating models is a good concept, there is also a danger that the metric and model chosen will lead to decreasing performance in future batches of data.

***Table 6.2:** Comparing the Outcomes of a Number of Drift Detection Methods With Different Models Using COVID-19 Data*



	Random Forest			Logistic Regression			Naïve Bayes		
	No Update	DDM	DDM OCI	No Update	DDM	DDM OCI	No Update	DDM	DDM OCI
Acc	84.34	87.50	85.31	85.77	88.20	88.00	83.59	86.30	83.12
Sens	87.28	81.80	85.43	67.29	80.90	84.16	77.51	73.32	80.20
Spec	84.11	87.70	85.30	87.24	88.40	88.12	83.65	86.64	83.20
Prec	16.92	17.20	14.94	17.02	18.00	18.32	15.05	16.26	11.87
F-Score	28.35	28.42	25.43	26.92	29.44	30.08	25.19	26.61	20.55

6.8 Conclusion

The importance of concept drift and its effects on healthcare AI models were investigated in this chapter, as well as the different types of concept drift. Additionally, this chapter presented the proposed methods (DDM and DDM-OCI) employed in my research to detect concept drift, as well as the datasets and experiments that were implemented. The findings of the DDM approach applied to the simulated, synthetic, COVID-19, and SSc datasets were also described and interpreted in this chapter. Detecting drift in data is crucial; drift detection methods ensure that a model will not become outdated over time when new data is generated. Therefore, this chapter explored how to correctly identify concept drift using standard drift detection methods. However, it is essential to understand the nature of drift in order to implement the appropriate performance metric. According to my findings, using an inaccurate metric can result in performance decline and an improper update. In addition, I discovered that DDM is sensitive to noise; therefore, it may detect apparent data drift caused by noise, which is not real drift. In a nutshell, the fundamental properties of healthcare might change over time, which could result in inaccurate predictions by AI models. Thus, the purpose of this chapter was to demonstrate the importance of monitoring AI models over time in order to prevent their obsolescence. Implementing drift detectors with a track record of success on my datasets allowed us to achieve this. In the future, however, concept drift could be addressed by monitoring my novel approaches in Chapters 4 and 5 using methods such as DDM and EDDM and adjusting these approaches to cluster and classify patient health outcomes over time. Alternative statistical methods that identify substantial differences in AI performance across two batches may also be utilised to detect drift and update the model accordingly. On the other hand, while these methods detect drift when the performance of a supervised learning model changes significantly over time, the target label (health outcomes) may be unknown or unavailable. In real-world applications, the true health outcomes may not be available until verified by experts, so there is a need for drift detection methods in unsupervised trials. Thus, concept drift detection techniques need to be addressed in order to handle unsupervised learning problems when true labels are not available. These techniques can determine whether the model's structure has changed over time and update it even in the absence of the true health outcome.

Chapter 7 Conclusion

▪ Introduction

The discovery of disease subtypes and the early detection of chronic diseases enhance patient survival and minimise healthcare expenses. Chronic disorders are now a top priority in the healthcare industry. As a result, healthcare systems across the world must provide patient care more effectively. In order to assist healthcare practitioners, enhance diagnosis, and provide individualised therapies, it would be useful to subtype uncommon diseases by producing prediction models that can recognise disease subclasses. In addition, ML could potentially play a vital role in forecasting complications of many uncommon diseases by identifying unmeasured effects and assisting in the development of a strong learning model. However, the underlying data may change over time, causing ML models to become outdated. In this chapter, I provide a summary of the approaches I employed to address and resolve the aforementioned issues. In addition, this chapter lays the groundwork for the work that I may conduct in the future.

▪ Achievements

The primary accomplishments of my thesis are depicted in Figure 7.1 below.

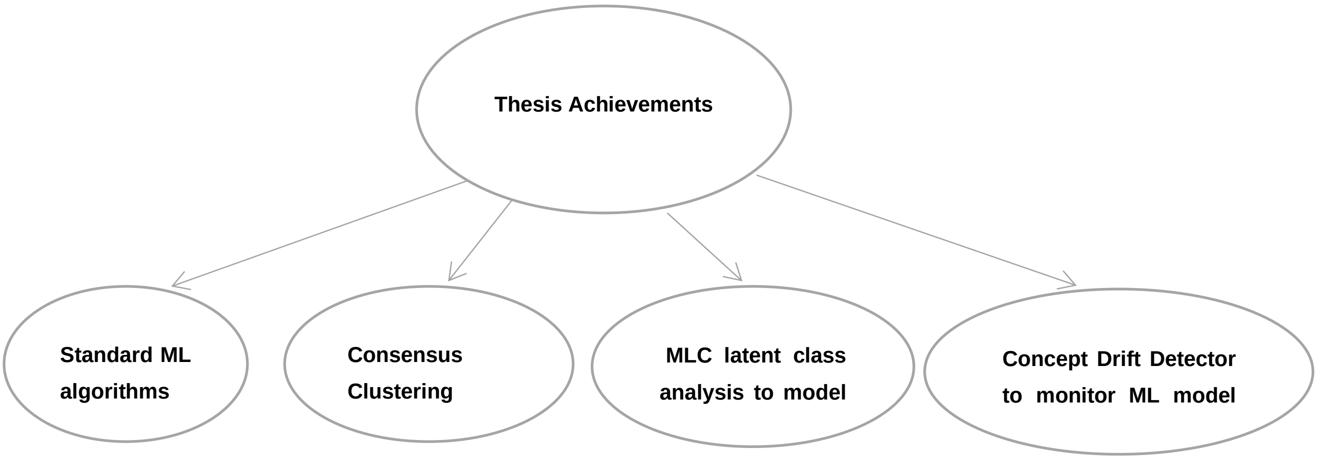

Thesis Achievements
Standard ML algorithms
Consensus Clustering
MLC latent class analysis to model
Concept Drift Detector to monitor ML model

Figure 7.1 Thesis Achievements

- **Exploration of state-of-the-art ML algorithms to model disease, with a focus on SSc.**

In order to make clinical outcome predictions and patient grouping, in this thesis I have applied several standard machine learning techniques to SSc datasets. These methods include decision tree (DT), naïve Bayes (NB), k-means, random forest, and latent class analysis algorithms. These methods have been analysed and evaluated using real-world datasets. I developed these algorithms to determine the impact of my suggested approaches relative to these algorithms.

- **Consensus clustering classification**

In this thesis, I introduced a new method that improves disease prediction and raises awareness of the disease's underlying features by combining consensus clustering techniques with classification methods. By utilising the consensus technique in conjunction with the C4.5 decision tree classifier (a transparent approach to classification that offers an understandable tree structure), my suggested method attempts to address both the inherent variation present in a number of different clustering approaches as well as the variance introduced by sample size. My proposed method therefore separates the data into training and test data. The training data is resampled to produce a set of consensus clusters. Then, each of these consensus clusters is utilised to construct a unique decision tree. Finally, using a single-linkage strategy, each test data point is classified in accordance with its distance from each identified consensus cluster. This is used to determine which decision tree is appropriate for classifying the data point. In other words, the agreement matrix is made by running k-means several times on different resampled training datasets. In order to create consensus cluster groups, I then apply the hierarchical clustering approach to this agreement matrix. A decision tree is constructed for each consensus group, and each test data point is categorised according to its closest distance from each consensus cluster. This method was applied to the SSc and breast cancer datasets. The results showed that my method outperforms all other techniques in terms of error rate and variance reduction. In addition, my method proved that clinical departments are able to apply this innovative approach to cluster patients and uncover essential traits in each group, which will allow for more reliable classification.

- **Multi-label classification (MLC) latent class analysis**

In this thesis, I have developed an ML model that provides a solution to the issue of comorbidities. Firstly, this model employs a latent class variable to cluster patients into groups by identifying hidden factors between observed variables. Secondly, in order to classify group labels jointly, this model uses a naïve Bayes MLC algorithm to predict various comorbidities. This algorithm is therefore comprised of two stages that occur simultaneously: LCA and MLC naïve Bayes. This model employs a 10-fold cross-validation procedure that separates the data into 90% training data and 10% test data. Latent class analysis is then used to group the training data. An MLC naïve Bayes model is built for each group. Each set of test data is assigned to one of the groups, and its labels are predicted based on the MLC naïve Bayes model within that group. There are two separate aspects of this model's performance. In the first, the model is compared to other common ML algorithms and evaluated with each label separately. For the second, it is compared to other common multi-label classification methods and evaluated with all labels jointly. The outcomes demonstrated that my model outperformed the competition, and it is a transparent method that could help deliver individualised patient care.

- **Concept drift alleviation**

The final accomplishment of this thesis is the implementation of DDM and DDM-OCI for batch data use. In my thesis, DDM monitors the overall error rate of the ML model over time. For every patient who arrives in a batch, error rate and standard deviation are calculated. According to my suggested approach, the model is not modified as long as the total error rate declines or remains constant. The model is retrained by adding data from the warning batch to the drift batch to the new model when the overall error rate reaches the drift level. On the other hand, the DDM-OCI method emphasises minority-class recall. Therefore, DDM-OCI checks the recall instead of the error rate. This method detects drift when the recall metric drops significantly. In my suggested approach, the model is not modified as long as the recall increases or remains constant. As before, the model is retrained by adding data from the warning batch to the drift batch to the new model when the recall reaches the drift level. Both methods have been used to monitor ML models over time using COVID-19, simulated, and SSc datasets. My results showed that methods for detecting drift are essential to assure that the model will not become outdated as new data are created. The results also showed that it is vital to understand the nature of the drift in order to implement the right performance metric.

- **Future works**

There are several limitations on this thesis, which will be explained in this section and may influence future research.

In Chapter 4, I presented the first novel model combining a consensus clustering method with classification to predict disease. The model is built on repeatedly executing k-means by resampling the training data to generate consensus groups. The resampling approach is one of the constraints of this thesis because different sampling techniques may produce varying outcomes. As a consequence, I may compare the outcomes of my model with other sampling techniques in the future. In addition, my model is built on repeatedly executing multiple k-means algorithms, which may be implemented using various clustering approaches but may generate a different number of clusters each time. In the future, I may consider running the model with alternative clustering methods than k-means each time and proposing a new strategy for locating the consensus groups among these methods. In addition, a weighted kappa approach may be used to assess the quality of the clustering method and compare the results of the resampled clustering approach. In this thesis, I grouped new patients using a single linkage in which each patient is clustered with the group to which they are geographically closest. I may investigate other approaches, such as average linkage and full linkage.

In Chapter 5, I presented another model combining the latent class method with MLC naïve Bayes to predict disease. As in Chapter 4, the resampling methodology is a limitation of this model, and thus I may examine alternative resampling approaches in future work. I have used the established EM algorithm to estimate the parameters for latent class analysis. Thus, it is necessary to look at new methods of estimating these parameters and compare them to the established EM algorithm. In addition, the number of clusters in my model is based on the Bayesian information criterion (BIC), and I might investigate different approaches. Using the relationship between the labels, my model predicts simultaneous complications using a modified naïve Bayes algorithm. However, I might consider adjusting various known transparent models, such as decision tree, to accommodate multi-label predictions. Both algorithms in Chapters 4 and 5 have been designed to predict disease by categorising patients into robust groups, which improves performance in disease prediction. However, I might consider expanding the method in Chapter 4 to address MLC and comparing it to the algorithm in Chapter 5.

In Chapter 6, I explored the significance of concept drift as well as the implementation of DDM and DDM-OCI for batch data use. Once again, the resampling strategy is a limitation because different sampling strategies may yield diverse results. I employed a strategy of under-sampling to balance the training dataset. Nonetheless, when over-sampling is utilised in place of any other sampling method, different results might be obtained. Another limitation of this thesis is that choosing the proper metric to monitor an AI model is crucial since different metrics might lead to different drifts in the model. Updating the model when drift is discovered is a crucial phase in which old and new data are added to the model, or the old data is discarded and only the new data is retained. Lastly, it is necessary to determine how I can monitor the model's performance when the real label values are not yet accessible. Therefore, I might investigate alternative sampling methods in the future in order to evaluate and contrast the effects of alternative sampling methods on concept drift outcomes. Furthermore, I might provide predictions for various labels and different concept drift detectors. In addition to forgetting or remembering previous data when updating the model, I may also examine other mechanisms for updating it, such as those that weight the significance of the data. Different concept drift strategies may be handled by monitoring my models in Chapters 4 and 5 over time and altering these approaches to clustering and categorising patient health outcomes in order to account for concept drift. Finally, in real-world applications, the true label is not always available. Thus, I might look at unsupervised concept drift detectors that detect drift of AI models before the true labels are available. I may present an unsupervised detector that identifies drift when the data distribution changes significantly over time. The benefit of this approach is that it does not require the true label to be available. This method might also uncover informative relationships between the variables; when these relationships change over time, it is an indication of drift.

References

Abozaid, G.M., Kerr, K., McKnight, A. and Al-Omar, H.A., 2022. Criteria to define rare diseases and orphan drugs: a systematic review protocol. BMJ open, 12(7), p.e062126.

Akhil Jabbar, M., Deekshatuluc, B. L., & Chandra, P. (2012). Heart disease prediction system using associative classification and genetic algorithm. International Conference on Emerging Trends in Electrical, Electronics and Communication Technologies-ICECIT, 2012, 183-192.

Al Abid,F., Mottalib,.M.(2012). An Accurate Grid based PAM Clustering Method for Large Dataset", International Journal of Computer Applications (0975 – 8887) Volume 41– No.21, March 2012.

Aldrees, A., Chikh, A., Berri, J. (2016). Comparative Evaluation of Four Multi-Label Classification Algorithms in Classifying Learning Objects. Computer Science Information Technology. 6. 10.5121/csit.2016.60210.

Alessandro, A., Corani, G., Mauá, D. and Gabaglio, S. (2013). An Ensemble of Bayesian Networks for Multilabel Classification. In: Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI). Beijing, China: pp.1220–1225.

Alexiou, A., Mantzavinos, V. D., Greig, N. H., & Kamal, M. A. (2017). A Bayesian model for the prediction and early diagnosis of Alzheimer's disease. Frontiers in Aging Neuroscience, 9, 77.

Altman, D.G. (1990). Practical Statistics for Medical Research (1st ed.). Chapman and Hall/CRC. https://doi.org/10.1201/9780429258589

Allanore, Y., & Distler, O. (2015). Advances in cohort enrichment shape future of trial design. Nature Reviews Rheumatology, 11(2), 72–74.

Bach, S., & Maloof, M. (2008). Paired Learners for Concept Drift. Proceedings - IEEE International Conference on Data Mining, ICDM. 23-32. 10.1109/ICDM.2008.119.

Baena-García, M.,Campo-Ávila, J.,Fidalgo-Merino, R.,Bifet, A., Gavald, R.,Morales, R. (2006). Early Drift Detection Method.

Bahl, N. (2017). Optimizing performance measures in classification using ensemble learning methods [Master's thesis, Arizona State University].

Balagatabi, Z. N., Balagatabi, H. N. (2013). Comparison of Decision Tree and SVM Methods in Classification of Researcher. Cognitive Styles in Academic Environment. Indian Journal of Automation and Artificial Intelligence, 1(1),31-43.

Beasley, W. B., & Rodgers, J. (2009). Resampling methods. In R. E. Millsap, & A. Maydeu-Olivares (eds.), The SAGE Handbook of Quantitative Methods in Psychology (362-386).

Becker, M. O., Distler, O., & Maurer, B. (2019). Systemische Sklerose – Klinisches Bild, Diagnostik und Therapie [Add the title translated in English]. Der Hautarzt, 70(9), 723–741.

Berrar, D. (2018). Cross-validation. Tokyo Institute of Technology.

Beyene, A.A., Welemariam, T., Persson, M. et al. (2015). Improved concept drift handling in surgery prediction and other applications. Knowl Inf Syst 44, 177–196

Bi, W., James, K. (2013). Efficient Multi-label Classification with Many Labels. Published in ICML 16 June 2013.

Bifet, A., Gavalda`, R.(2007).Learning from time-changing data with adaptive windowing. In: Proceedings of the seventh SIAM international conference on data mining. 26–28 Apr 2007, Minneapolis. p. 443–448.

Black, N., Martineau, F., & Manacorda, T. (2015). Diagnostic odyssey for rare diseases: Exploration of potential indicators. Policy Innovation Research Unit.

Blumenthal, D. (2010). Launching HITECH. The New England Journal of Medicine, 362(5), 382–385. https://doi.org/10.1056/NEJMp0912825

Boban, B. R., Cillamol, K. J., Cheruvil, E., Thomas, S., & Abraham, T. (2019). Hypokalemic periodic paralysis: A case report. International Journal of Trend in Scientific Research and Development, 3(3), 216–217.

Bohra, H., Arora, A., Gaikwad, P., Bhand, R., & Patil, M. (2017). Health prediction and medical diagnosis using Naive Bayes. International Journal of Advanced Research in Computer and Communication Engineering, 4(6), 32-35.

Bonomi, F., Peretti, S., Lepri, G., Venerito, V., Russo, E., Bruni, C., Iannone, F., Tangaro, S.S., Amedei, A., Guiducci, S., Matucci Cerinic, M., & Bellando Randone, S. (2022). The Use and Utility of Machine Learning in Achieving Precision Medicine in Systemic Sclerosis: A Narrative Review. *Journal of Personalized Medicine, 12*.

Borkar, A. R., & Deshmukh, P. R. (2015). Naïve bayes classifier for prediction of swine flu disease. International Journal of Advanced Research in Computer Science and Software Engineering, 5(4), 120–123.

Bosoni, P. (2016). Discovery of disease subclasses by combining supervised and unsupervised learning [master's thesis]. University of Pavia.

Bosoni, P., Nihtyanova, S., Denton ,C., Tucker ,A.(2016).Combining Unsupervised and Supervised Learning for discovering Disease Subclasses. 225-226. 10.1109/CBMS.2016.37.

Brown, G. (2010). Ensemble learning. In C. Sammut, & G. I. Webb (eds.), Encyclopedia of Machine Learning (312-320). Springer.

Bruni, C., Cuomo, G., Rossi, F. W., Praino, E., & Bellando-Randone, S. (2018). Kidney involvement in systemic sclerosis: From pathogenesis to treatment. Journal of Scleroderma and Related Disorders, 3(1), 43–52.

Chakraborty, T. (2017). EC3: Combining clustering and classification for ensemble learning. IEEE International Conference on Data Mining, 2017, 781-786. https://doi.org/10.1109/ICDM.2017.92

Champion, H. C. (2008). The heart in scleroderma. Rheumatic Disease Clinics of North America, 34(1), 181–190.

Chapron, K., Plantevin, V., Thullier, F., Bouchard, K., Duchesne, E., & Gaboury, S. (2018). A more efficient transportable and scalable system for real-time activities and exercises recognition. Sensors, 18(1), Article 268.

Chaurasia, V., Saurabh, P., & Tiwari, BB. (2018). Prediction of benign and malignant breast cancer using data mining techniques. Journal of Algorithms & Computational Technology, 12(2), 119-126.

Chernbumroong, S., Johnson, J., Gupta, N., Miller, S., McCormack, F. X., Garibaldi, J. M., & Johnson, S. R. (2020). Machine learning can predict disease manifestations and outcomes in lymphangioleiomyomatosis. European Respiratory Journal, 57, 2003036.

Correa da Silva, F. S., Costa, F., & Lemma, A. (2020). Practical considerations regarding classification learning for clinical diagnosis and therapy advice in oncology. ICT Express. https://doi.org.6.10.1016/j.icte.2020.03.004

Cristianini, N., and Shawe-Taylor, J.(2000). An Introduction to Support Vector Machines: And Other Kernel-based Learning Methods. Cambridge University Press, 1st.

Cutler, A., Cutler, D. R., & Stevens, J. R. (2012). Random forests. In C. Zhang & Y. Ma (Eds.), Ensemble machine learning: Methods and applications (pp. 157–176). Springer.

De Grandis, G. and Halgunset, V., 2016. Conceptual and terminological confusion around personalised medicine: a coping strategy. *BMC Medical Ethics*, *17*(1), pp.1-12.

Denton, C. P., & Khanna, D. (2017). Systemic sclerosis. The Lancet, 390(10103), 1685–1699.

Deo, R. C. (2015). Machine learning in medicine. Circulation, 132(20), 1920-1930. https://doi.org/10.1161/CIRCULATIONAHA.115.001593

Dharmadhikari, S.C., Ingle, M. and Kulkarni, P. (2012). Learning Deep Latent Spaces for Multi Label Classification. International Journal of Advanced Computer Science and Applications (IJACSA), USA

Dhongade, P., Longadge, R. and Kapgate, D. (2014) A Review on Classification of Multi-label Data in Data Mining, International Journal of Computer Science and Mobile Computing, Vol.3(12).

Dodangeh, E., Choubin, B., Eigdir, A. N., Nabipour, N., Panahi, M., Shamshirband, S., & Mosavi, A. (2020). Integrated machine learning methods with resampling algorithms for flood susceptibility prediction. Science of the Total Environment, 705, 135983.

Duckworth, C., Chmiel, F. P., Burns, D. K., Zlatev, Z. D., White, N. M., Daniels, T., Kiuber, M., & Boniface, M. J. (2021). Using explainable machine learning to characterise data drift and detect emergent health risks for emergency department admissions during COVID-19. Scientific Reports, 11(1), 23017.

Elton, J., & O'Riordan, A. (2016). Healthcare disrupted: Next generation business models and strategies. Wiley.

Eshghi, M., Haughton, D., Legrand, P., Skaletsky, M. and Woolford, S. (2011). Identifying Groups: A Comparison of Methodologies. Journal of Data Science, 9, pp.271–291.

Falavigna, G., Costantino, G., Furlan, R., Quinn, J. V., Ungar, A., & Ippoliti, R. (2019). Artificial neural networks and risk stratification in emergency departments. Internal and Emergency Medicine, 14(2), 291-299.

Feng G. (2019). CONCEPT DRIFT DETECTION FOR MACHINE LEARNING WITH STREAM DATA. A thesis submitted for the Degree of Doctor of Philosophy. Faculty of Engineering and Information Technology University of Technology Sydney.2019 Clinical Practice Research.

Forbes, A. and Marie, I. (2009). Gastrointestinal complications: the most frequent internal complications of systemic sclerosis. Rheumatology, 48(3):iii36–iii39.

Gadewadikar, J., Kuljaca, O., Agyepong, K., Sarigul, E., Zheng, Y, & Zhang, P. (2010). Exploring Bayesian Networks for medical decision support in breast cancer detection. African Journal of Mathematics and Computer Science Research, 3(10), 225-231. http://www.academicjournals.org/AJMCSR

Gama, J., Indrė, Ž., Albert, B., Mykola, P., Bouchachia,A. (2014).A survey on concept drift adaptation. ACM Computing Surveys (CSUR).

Gammie, T., Lu, C. Y., & Babar, Z. U. (2015). Access to orphan drugs: A comprehensive review of legislations, regulations and policies in 35 Countries. PLoS One, 10(10), 1-24.

Giorgio, C., Mauro, S. (2016).Air pollution prediction via MLC, Environmental Modelling.Volume 80,2016,Pages 259-264,ISSN 1364-8152

Goder, A,. and Filkov, S.(2008). Consensus Clustering Algorithms: Comparison and Refine- ment. In Munro J.I and Wagner D., editors, 2008 Proceedings of the Tenth Workshop on Algorithm Engineering and Experiments (ALENEX).

Goecks, J., Jalili, V., Heiser, L. M., & Gray, J. W. (2020). How machine learning will transform biomedicine. Cell, 181(1), 92–101.

Goel M,K,. and Khanna, P,. Kishore, J.(2010). Understanding survival analysis:Kaplan- Meier estimate. International Journal of Ayurveda Research,1(4):274–278, October.

Gonçalves Jr, P., & Santos, S., & Barros, R., & Vieira, D. (2014). A Comparative Study on Concept Drift Detectors. Expert Systems with Applications. 41. 8144-8156. 10.1016/j.eswa.2014.07.019.

Gonçalves, L., Subtil, A., de, O., Rosário, V., Lee, P., Shaio, M. (2012). Bayesian Latent Class Models in malaria diagnosis. PMC3402519.

Goundry, B., Bell, L., Langtree, M., & Moorthy, A. (2012). Diagnosis and management of Raynaud’s phenomenon. BMJ, 344, Article e289.

Guan, W. J., Jiang, M., Gao, Y. H., Li, H. M., Xu, G., Zheng, J. P., Chen, R. C., & Zhong, N. S. (2016). Unsupervised learning technique identifies bronchiectasis phenotypes with distinct clinical characteristics. The International Journal of Tuberculosis and Lung Disease, 20(3), 402-410.

Gupta, A., Kumar, L., Jain, R., Nagrath, P. (2020). Heart disease prediction using classification (Naive Bayes). In P. Singh, W. Pawłowski, S. Tanwar, N. Kumar, J. Rodrigues, & M. Obaidat (eds.), Proceedings of first International Conference on Computing, Communications, and Cyber-Security (IC4S 2019) (561-573). Springer. https://doi.org/10.1007/978-981-15-3369-3_42

Hachulla, E., & Launay, D. (2011). Diagnosis and classification of systemic sclerosis. Clinical Reviews in Allergy & Immunology, 40(2), 78–83.

Halkidi, M., Batistakis, Y., & Vazirgiannis, M. (2001). On clustering validation techniques. Journal of Intelligent Information Systems, 17(2–3), 107–145.

Hamish,H., Yun Sing, K., Gillian, D., Edmond,Z. (2020).Detecting Concept Drift In Medical Triage. Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval.Association for Computing Machinery, New York, NY, USA, 1733–1736.

Handelman, G. S., Kok, H. K., Chandra, R. V., Razavi, A. H., Lee, M. J., & Asadi, H. (2018). eDoctor: Machine learning and the future of medicine. Journal of Internal Medicine, 284(6), 603–619. https://doi.org.10.1111/joim.12822

Hashmani, M. A., Syed, M. J., Rehman, M., & Inoue, A. (2020). Concept Drift evolution in machine learning approaches: A systematic literature review. International Journal on Smart Sensing and Intelligent Systems, 13, 1-16.

Herrick, A. L. (2018). Systemic sclerosis: Clinical features and management. Medicine, 46(2), 131–139.

Jagga, Z. and Gupta, D. (2014) Supervised Learning Classification Models for Prediction of Plant Virus Encoded RNA Silencing Suppressors. Available at: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0097446.

Jain, A. K., Duin, R. P. W., & Mao, J. (2000). Statistical pattern recognition: A review. IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(1), 4-37.

Jaramillo-Valbuena, S., Londoño-Peláez, J. M., & Augusto Cardona, S. (2017). Performance evaluation of concept drift detection techniques in the presence of noise. Espacios, 38(39), Article 16.

Johnson, K. B., Wei, W. Q., Weeraratne, D., Frisse, M. E., Misulis, K., Rhee, K., Zhao, J., & Snowdon, J. L. (2020). Precision Medicine, AI, and the future of personalized health care. Clinical and Translational Science, 14(1), 86-93.

Kabir, M.,Rahman, C., Hossain, M., Dahal, K. (2011). Enhanced Classification Accuracy on Naive Bayes Data Mining Models. International Journal of Computer Applications. 28. 9-16. 10.5120/3371-4657.

Kajan, S., Pernecký, D., & Goga, J. (2014). Application of neural network in medical diagnostics. Slovak University of Technology in Bratislava.

Kalra, D., (2019). The importance of real-world data to precision medicine. *Personalized Medicine*, *16*(2), pp.79-82.

Kalyani, P. (2012). Approaches to Partition Medical Data using Clustering Algorithms. International Journal of Computer Applications, V49,N23, P7-10,2012.

Kayser, C., & Fritzler, M. J. (2015). Autoantibodies in systemic sclerosis: Unanswered questions. Frontiers in Immunology, 6, Article 167.

Kellam, P., & Liu, X., & Martin, N., & Orengo, C., & Swift, S., & Tucker, A. (2001). Comparing, contrasting and combining clusters in viral gene expression data. Proceedings of the IDAMAP2001 Workshop.

Kharya, S., Agrawal, S., & Soni, S. (2014). Naive Bayes classifiers: A probabilistic detection model for breast cancer. International Journal of Computer Applications, 92(10), 26-31.

Kim, J., Lee, J., & Lee, Y. (2015). Data-mining-based coronary heart disease risk prediction model using fuzzy logic and decision tree. Healthcare Informatics Research, 21(3), 167-174.

König, I. R., Fuchs, O., Hansen, G., von Mutius, E., & Kopp, M. V. (2017). What is precision medicine? European Respiratory Journal, 50(4).

Kotsiantis, S. B., Zaharakis, I., & Pintelas, P. (2007). Supervised machine learning: A review of classification techniques. Frontiers in Artificial Intelligence and Applications, 160, 3-24.

Kovalenko, O. (2020). Machine learning: 12 real-world applications of machine learning in healthcare. SPD Group. https://spd.group/machine-learning/machine-learning-in-healthcare/

Lanza, S., Rhoades, BL. (2013). Latent class analysis: an alternative perspective on subgroup analysis in prevention and treatment. PMID: 21318625; PMCID: PMC3173585.

Law, H., Harrington, R. (2016). A Primer on Latent Class Analysis. Department of Pharmacy Systems. University of Illinois at Chicago, Chicago, Illinois, USA

Lee, P. H. (2014). Resampling methods improve the predictive power of modeling in class-imbalanced datasets. *International Journal of Environmental Research and Public Health,* 11(9), 9776-9789.

Liu,G., Li, G., Wang, Y., Wang Y.(2010). Modelling of inquiry diagnosis for coronary heart disease in Traditional Chinese Medicine by using multilabel learning. BMC Complement Altern Med. 2010 Jul 20;10:37. doi: 10.1186/1472-6882-10-37. PMID: 20642856; PMCID: PMC2921356.

Liu, H., Zhao, R., Fang, H., Cheng, F., Fu, Y. and Liu, Y.-Y. (2017). Entropy-based consensus clustering for patient stratification. Bioinformatics, 33(17), pp.2691–2698. doi:10.1093/bioinformatics/btx167.

Lo, H. Y., Chang, C. M., Chiang, T. H., Hsiao, C. Y., Huang, A., Kuo, T. T., Lai, W. C., Yang, M. H., Yeh, J. J., Yen, C. C., & Lin, S. D. (2008). Learning to improve area-under-FROC for imbalanced medical data classification using an ensemble method. ACM SIGKDD Explorations Newsletter, 10(2), 43-46.

Lopez, C., Tucker, S., Salameh, T., & Tucker, C. (2018). An unsupervised machine learning method for discovering patient clusters based on genetic signatures. Journal of Biomedical Informatics, 85, 30-39.

Lourenço, A, Carreiras, C., Bulò, S. R., & Fred, A. (2014). ECG analysis using consensus clustering. 22nd European Signal Processing Conference, 511-515.

Lu, J., & Liu, A., & Dong, F., & Gu, F., & Gama, J., & Zhang, G. (2018). Learning under Concept Drift: A Review. IEEE Transactions on Knowledge and Data Engineering. PP. 1-1. 10.1109/TKDE.2018.2876857.

Lucas, P. J. (2001). Bayesian Networks in medicine: A model-based approach to medical decision making. Unite. https://www.cs.ru.nl/P.Lucas/eunite.pdf

Lucas, P. J., Van-der-Gaag, L. C., & Abu-Hanna, A. (2004). Bayesian networks in biomedicine and healthcare. Artificial Intelligence in Medicine, 30(3),201-214.

Lütz, E. (2019). Unsupervised learning to detect patient subgroups in electronic health records. KTH Royal Institute of Technology School of Electrical Engineering and Computer Science.

MacLeod, H., Yang, S., Oakes, K., Connelly, K., & Natarajan, S. (2016). Identifying rare diseases from behavioural data: A machine learning approach. 2016 IEEE First International Conference on Connected Health: Applications, Systems and Engineering Technologies, 130-139.

Madadipouya, K. (2015). A new decision tree method for data mining in medicine. Advanced Computational Intelligence: An International Journal, 2(3), 31-37.

Magidson, J.,Vermunt, J. K. (2005). A nontechnical introduction to latent class models. DMA Research Council Journal.

Masoudnia, S., & Ebrahimpour, R. (2014). Mixture of experts: A literature survey. Artificial Intelligence Review, 42, 275-293.

Mooijaart, A., Heijden,M.(1992). The EM algorithm for latent class analysis with equality constraints. Psychometrika. 57. 261-269.10.1007/BF02294508.

Mori, M., Krumholz, H. M., & Allore, H. G. (2020). Using latent class analysis to identify hidden clinical phenotypes. *JAMA*, 324(7), 700–701.

Müller, M. & Salathé, M. (2020). Addressing machine learning concept drift reveals declining vaccine sentiment during the COVID-19 pandemic. Computer Science.

Nareshpalsingh, M.J,. and Modi, N.H. (2017). Multi-Label Classification Methods: A Comparative Study. International Research Journal of Engineering and Technology (IRJET), 4(12), pp. 263-270.

Nguyen, N., & Caruana, R. (2007). Consensus Clusterings. *Seventh IEEE International Conference on Data Mining (ICDM 2007)*, 607-612.

Nielsen, F. (2016). Hierarchical Clustering. 10.1007/978-3-319-21903-5_8

Nimmesgern, E., Benediktsson, I., & Norstedt, I. (2017). Personalized medicine in Europe. Clinical and Translational Science, 10(2), 61-63.

Nithya, N., Duraiswamy, K., & Gomathy, P. (2014). A survey on clustering techniques in medical diagnosis. International Journal of Computer Science Trends and Technology, 1(2), 17-23.

Ogbuabor, G. & Ugwoke, F. (2018). Clustering algorithm for a healthcare dataset using silhouette score value. International Journal of Computer Science & Information Technology, 10(2), 27-37.

Pandey, A. K., Pandey, P., Jaiswal, K., & Sen, A. K. (2013). Datamining clustering techniques in the prediction of heart disease using attribute selection method. Heart Disease, 14, 16-17.

Panopoulos, S., Tektonidou, M., Drosos, A. A., Liossis, S. N., Dimitroulas, T., Garyfallos, A., Sakkas, L., Boumpas, D., Voulgari, P. V., Daoussis, D., Thomas, K., Georgiopoulos, G., Vosvotekas, G., Vassilopoulos, D., & Sfikakis, P. P. (2018). Prevalence of comorbidities in systemic sclerosis versus rheumatoid arthritis: A comparative, multicenter, matched-cohort study. Arthritis Research & Therapy, 20, 267.

Pastorino, R., De Vito, C., Migliara, G., Glocker, K., Binenbaum, I., Ricciardi, W., & Boccia, S. (2019). Benefits and challenges of Big Data in healthcare: An overview of the European initiatives. European Journal of Public Health, 3(29), 23-27.

Peberdy, V. (2017). Rare Diseases: Shaping a future with no-one left behind. IFPMA.

Pelter, M.N. and Druz, R.S., 2022. Precision Medicine: Hype or Hope? Trends in Cardiovascular Medicine.

Podgorelec, V., Kokol, P., Stiglic, B., & Rozman, I. (2002). Decision trees: An overview and their use in medicine. Journal of Medical Systems, 26(5), 445-463.

Pottel, H. (2015). Resampling methods in Excel.

Prajapati, P., Thakkar, A., & Ganatra, A.P. (2012). A Survey and Current Research Challenges in Multi-Label Classification Methods.

Praveena, M.,Jaiganesh, V. (2017). A literature review on supervised machine learning algorithms and boosting process. International Journal of Computer Applications, 169(8), 32-35.

Rajalakshmi, K., Dhenakaran, D. S., & Roobin, N. (2015). Comparative analysis of K-means algorithm in disease prediction. International Journal of Science, Engineering and Technology Research, 4(7).

Ratnawati, D. E., Priandani, N. D. and Machsus, M. (2018). A modified K-means with Naïve Bayes (KMNB) algorithm for breast cancer classification. Journal of Telecommunication, Electronic and Computer Engineering, 10(6), 137-140.

Read, J., Pfahringer, B., Holmes, G. (2011). Classifier chains for MLC. Mach Learn 85,333 (2011).

Refaeilzadeh, P., Tang, L., & Liu, H. (2009). Cross-validation. In L. Liu & M. T. Özsu (Eds.), Encyclopedia of database systems (2009 ed.). Springer.

Ren, Z., & Wang, C. (2018). Application of dynamic clustering algorithm in medical surveillance. Computer Science and Information Technology, 83-86.

Richter, T., Nestler-Parr, S., Babela, R., Khan, Z.M., Tesoro, T., Molsen, E. and Hughes, D.A., (2015). Rare disease terminology and definitions—a systematic global review: report of the ISPOR rare disease special interest group. Value in health, 18(6), pp.906-914.

Rodwell, C., & Aymé, S. (2015). Rare disease policies to improve care for patients in Europe. Biochimica et Biophysica Acta (BBA) – Molecular Basis of Disease, 1852(10), 2329-2335.

Saarela, M., Ryynänen, O. P., & Äyrämö, S. (2019). Predicting hospital associated disability from imbalanced data using supervised learning. Artificial Intelligence in Medicine, 95, 88-95.

Santos, A., Neto, A. (2011). A Comparative Analysis of Classification Methods to Multi-label Tasks in Different Application Domains. International Journal of Computer Information Systems and Industrial Management Applications.

Schaefer, J., Lehne, M., Schepers, J., Prasser, F., & Thun, S. (2020). The use of machine learning in rare diseases: A scoping review. Orphanet Jourlnal of Rare Diseases, 15(1), 145.

Schleidgen, S., Klingler, C., Bertram, T., Rogowski, W.H. and Marckmann, G., (2013). What is personalized medicine: sharpening a vague term based on a systematic literature review. BMC medical ethics, 14(1), pp.1-12.

Schork, N. J. (2019). Artificial Intelligence and Personalized Medicine. In D. von Hoff, & H. Han (eds.), Precision Medicine in Cancer Therapy (265-283). Springer.

Shi, X., Qu, T., Van Pottelbergh, G., van den Akker, M., & De Moor, B. (2022). A resampling method to improve the prognostic model of end-stage kidney disease: A better strategy for imbalanced data. Frontiers in Medicine, 9, 730748.

Shouman, M., Turner, T.and Stocker, R. (2012). Using Data Mining Techniques in Heart Disease Diagnosis and Treatment. In: Proceedings of IEEE Japan-Egypt Conference on Electronics Communications and Computers, vol. 2, pp. 174-177.

Sidey-Gibbons, J. A., & Sidey-Gibbons, C. J. (2019). Machine learning in medicine: A practical introduction. BMC Medical Research Methodology, 19(1), 64. https://doi.org/10.1186/s12874-019-0681-4

Sisodia, D., & Sisodia, D. S. (2018). Prediction of diabetes using classification algorithms, Procedia Computer Science, 132, 1578-1585. https://doi.org/10.1016/j.procs.2018.05.122

Soni, H., Vyas, A., & Singh, U. (2018). Identify rare disease patients from electronic health records through machine learning approach. 2018 International Conference on Inventive Research in Computing Applications, 1390-1395.

Soni, J., Ansari, U. (2011). Predictive Data Mining Diagnosis: An overview of Heart Disease Prediction. Internation Journal(0975-8887), val 17,N 8,2011.

Srinivas, P., Bhattacharyya, D., & Chakkaravarthy, D. M. (2020). An artificial intelligent based system for efficient swine flu prediction using Naive Bayesian classifier.

International Journal of Current Research and Review, 12, 134-139. https://doi.org/10.31782/IJCRR.2020.121519

Steen, V. D. (2005). Autoantibodies in systemic sclerosis. Seminars in Arthritis and Rheumatism, 35(1), 35–42.

Sun, M., Sun, X., Shan, D.(2019). Pedestrian crash analysis with latent class clustering method. PMID: 30623856.

Svenstrup, D., Jørgensen, H. L., & Winther, O. (2015). Rare disease diagnosis: A review of web search, social media and large-scale data-mining approaches. Rare diseases (Austin, Tex.), 3(1), e1083145

Swift, S., Tucker, A., Vinciotti, V., Martin, N., Orengo, C., Liu, X., Kellam, P. (2004).Consensus clustering and functional interpretation of gene-expression data.Genome Biol. 5(11):R94. doi: 10.1186/gb-2004-5-11-r94. Epub 2004 Nov 1. PMID: 15535870; PMCID: PMC545785.

Takada, M., Sugimoto, M., Naito, Y., Moon, H. G., Han, W., Noh, D. Y., Kondo, M., Kuroi, K., Sasano, H., Tomita, M., & Toi, M. (2012). Prediction of axillary lymph node metastasis in primary breast cancer patients using a decision tree-based model. BMC Medical Informatics and Decision Making, 12(1), 54. https://doi.org/10.1186/1472-6947-12-54

Tan, P. N., Steinbach, M., & Kumar, V. (2006). Cluster analysis: Basic concepts and algorithms. Introduction to Data Mining, 8, 487-568.

Topaloglu, M., & Malkoç G. (2016). Decision tree application for renal calculi diagnosis. International Journal of Applied Mathematics, Electronics and Computers, Special Issue 1, 404-407. https://doi.org/10.18100/ijamec.281134

Tsoumakas, G., Katakis, I. (2009). Multi-Label Classification: An Overview. International Journal of Data Warehousing and Mining. 3. 1-13. 10.4018/jdwm.2007070101.

Tucker, A., Garway, D. (2010). The Pseudotemporal Bootstrap for Predicting Glaucoma from Cross-Sectional Visual Field Data. IEEE, Vol 14, N1,2010.

Tucker, A., Wang, Z., Rotalinti, Y., & Myles, P. (2020). Generating high-fidelity synthetic patient data for assessing machine learning healthcare software. NPJ Digital Medicine, 3(1), 147. https://doi.org.10.1038/s41746-020-00353-9

Tyndall, A. J., Bannert, B., Vonk, M., Airò, P., Cozzi, F., Carreira, P. E., Bancel, D. F., Allanore, Y., Müller-Ladner, U., Distler, O., Iannone, F., Pellerito, R., Pileckyte, M., Miniati, I., Ananieva, L., Gurman, A. B., Damjanov, N., Mueller, A., Valentini, G. (2010). Causes and risk factors for death in systemic sclerosis: A study from the EULAR scleroderma trials and research (EUSTAR) database. Annals of the Rheumatic Diseases, 69(10), 1809–1815.

Vayena, E., Blasimme, A., & Cohen, I. G. (2018). Machine learning in medicine: Addressing ethical challenges. PLoS Medicine, 15(11).

Vega-Pons, S., & Ruiz-Shulcloper, J. (2011). A Survey of Clustering Ensemble Algorithms. Int. J. Pattern Recognit. Artif. Intell., 25, 337-372.

Vembandasamy, K., Sasipriya, R., Deepa, E. (2015). Heart Diseases Detection Using Naive Bayes Algorithm. International Journal of Innovative Science, Engineering &amp; Technology, 2 (9), pp. 441-444.

Vicente, A. M., Ballensiefen, W., & Jonsson, J. I. (2020). How personalized medicine will transform healthcare by 2030: The ICPerMed vision. Journal of Translational Medicine, 18, 180. https://doi.org/10.1186/s12967-020-02316-w

Wang, S., Minku, L., Ghezzi, D., Caltabiano, D., Tino, P., Yao, X.(2013). Concept drift detection for online class imbalance learning, The 2013 International Joint Conference on Neural Networks (IJCNN), 1–10, (2013) https://doi.org/doi: 10.1109/IJCNN.2013.6706768

Webb, G., & Lee, L., & Petitjean, F., & Goethals, B. (2017). Understanding Concept Drift.

Wei, W., Visweswaran, S., & Cooper, G. F. (2011). The application of Naive Bayes model averaging to predict Alzheimer’s disease from genome-wide data. Journal of the American Medical Informatics Association, 18(4), 370-375. https://doi.org/10.1136/amiajnl-2011-000101

Wei, Z., Zhang, H., Zhang, Z., Li, W., & Miao, D. (2011). A Naive Bayesian Multi-label Classication Algorithm With Application to Visualize Text Search Results.

Wiens, J., & Wallace, B. C. (2016). Editorial: Special issue on machine learning for health and medicine. Machine Learning, 102, 305–307. https://doi.org/10.1007/s10994-015-5533-9

Wu, P., Liu, J., Pei, S., Wu, C., Yang, K., Wang, S. and Wu, S. (2018). Integrated genomic analysis identifies clinically relevant subtypes of renal clear cell. carcinoma. BMC Cancer, 18(1), 2018.

www.raredisease.org.uk

Xiao, G,. and Pan, W.(2007). Consensus Clustering of Gene Expression Data and Its Applica- tion to Gene Function Prediction. Journal of Computational and Graphical Statistics, 16(3):733–751, September 2007.

Yaghi, S., Novikov, A., & Trandafirescu, T. (2020). Clinical update on pulmonary hypertension. Journal of Investigative Medicine, 68(4), 821–827.

Ying, Y., Witold P., Duoqian D. (2014). Multi-label classification by exploiting label correlations, Expert Systems with Applications, Volume 41, Issue 6,2014, Pages 2989- 3004,ISSN 0957-4174

Yongwen, Jiang., Dora, M., Dumont, T., Kristi, P., Samara, B. (2015).A latent class model to identify city/town chronic disease patterns, Preventive Medicine, Volume 73,2015,Pages 139-144,ISSN 0091-7435.

Yousefi, L., Swift, S., Arzoky, M., Saachi, L., Chiovato, L., & Tucker, A. (2018). Opening the black box: Discovering and explaining hidden variables in type 2 diabetic patient modelling. IEEE International Conference on Bioinformatics and Biomedicine, 2018, 1040-1044.

Zhou, L., Yin, D., Zheng,Y.,Li, Y. (2018). Traditional Chinese Medicine (TCM) Diagnosis Model BuildingBased on Multi-label Classification. MATEC Web of Conferences. 232. 02026.

Zhu, P., Zhu, W., Hu, Q., Zhang, C. and Zuo, W. (2017). Subspace clustering guided unsupervised feature selection. Pattern Recognition, 66, pp.364-374.

Zufferey, D., Hofer, T., Hennebert, J., Schumacher, M., Ingold, R., Bromuri, S.(2015). Performance comparison of multilabel learning algorithms on clinical data for chronic diseases. Comput Biol Med. 2015 Oct 1;65:34-43.