CONTRIBUTIONS TO EVALUATION OF MACHINE LEARNING MODELS

O.A.M. RADO

PHD

UNIVERSITY OF BRADFORD

2019

Contributions to evaluation of machine learning models
Applicability domain of classification models

Omesaad Abobaker Mohamed RADO

Submitted for the Degree of

Doctor of Philosophy

Faculty of Engineering and Informatics

University of Bradford

2019

Abstract

Omesaad Abobaker Mohamed RADO

Contributions to evaluation of machine learning models

The applicability domain of classification models

Keywords: Machine learning, Classification algorithms, Binary classification, Accuracy, Model evaluation, Model reliability, Applicability domain, Model robustness, Model coverage, Healthcare data.

Artificial intelligence (AI) and machine learning (ML) present some application opportunities and challenges that can be framed as learning problems. The performance of machine learning models depends on algorithms and the data. Moreover, learning algorithms create a model of reality through learning and testing with data processes, and their performance shows an agreement degree of their assumed model with reality. ML algorithms have been successfully used in numerous classification problems. With the developing popularity of using ML models for many purposes in different domains, the validation of such predictive models is currently required more formally. Traditionally, there are many studies related to model evaluation, robustness, reliability, and the quality of the data and the data-driven models. However, those studies do not consider the concept of the applicability domain (AD) yet. The issue is that the AD is not often well defined, or it is not defined at all in many fields. This work investigates the robustness of ML classification models from the applicability domain perspective. A standard definition of applicability domain regards the spaces in which the model provides results with specific reliability.

The main aim of this study is to investigate the connection between the applicability domain approach and the classification model performance. We are examining the usefulness of assessing the AD for the classification model, i.e. reliability, reuse, robustness of classifiers. The work is implemented using three approaches, and these approaches are conducted in three various attempts: firstly, assessing the applicability domain for the classification model; secondly, investigating the robustness of the classification model based on the applicability domain approach; thirdly, selecting an optimal model using Pareto optimality. The experiments in this work are illustrated by considering different machine learning algorithms for binary and multi-class classifications for healthcare datasets from public

benchmark data repositories. In the first approach, the decision trees algorithm (DT) is used for the classification of data in the classification stage. The feature selection method is applied to choose features for classification. The obtained classifiers are used in the third approach for selection of models using Pareto optimality. The second approach is implemented using three steps; namely, building classification model; generating synthetic data; and evaluating the obtained results.

The results obtained from the study provide an understanding of how the proposed approach can help to define the model's robustness and the applicability domain, for providing reliable outputs. These approaches open opportunities for classification data and model management. The proposed algorithms are implemented through a set of experiments on classification accuracy of instances, which fall in the domain of the model. For the first approach, by considering all the features, the highest accuracy obtained is 0.98, with thresholds average of 0.34 for Breast cancer dataset. After applying recursive feature elimination (RFE) method, the accuracy is 0.96% with 0.27 thresholds average. For the robustness of the classification model based on the applicability domain approach, the minimum accuracy is 0.62% for Indian Liver Patient data at r=0.10, and the maximum accuracy is 0.99% for Thyroid dataset at r=0.10. For the selection of an optimal model using Pareto optimality, the optimally selected classifier gives the accuracy of 0.94% with 0.35 thresholds average.

This research investigates critical aspects of the applicability domain as related to the robustness of classification ML algorithms. However, the performance of machine learning techniques depends on the degree of reliable predictions of the model. In the literature, the robustness of the ML model can be defined as the ability of the model to provide the testing error close to the training error. Moreover, the properties can describe the stability of the model performance when being tested on the new datasets. Concluding, this thesis introduced the concept of applicability domain for classifiers and tested the use of this concept with some case studies on health-related public benchmark datasets.

Publications and presentations

1. Publications

- O. Rado, N. Ali, H. M. Sani, A. Idris, and D. Neagu, “Performance Analysis of Feature Selection Methods for Classification of Healthcare Datasets,” Springer”, Cham, 2019, pp. 929–938.

- O. Rado, M. Al Fanah, and E. Taktek, “Ensemble of Multiple Classification Algorithms to Predict Stroke Dataset, ,” Springer”, Cham ,2019, pp. 93–98.

- O. Rado, M. Al Fanah, and E. Taktek, “Performance Analysis of Missing Values Imputation Methods Using Machine Learning Techniques,” Springer”, Cham, 2019, pp. 738–750.

- Accpted paper " On Selection of Optimal Classifiers " in AI-2019 Thirty-ninth SGAI International Conference on Artificial Intelligence. Cambridge, England 17-19 December 2019.

- O.Rado, " On Selection of Optimal Classifiers " , Annual Innovative Engineering Research Conference (AIERC 2019), University of Bradford, 15.07.2019.

- Submitted (under review) a paper titled " Robustness of machine learning models based on applicability domain approach, " International Journal of Expert Systems, Wiley, 22.07.2019.

2. Presentations

- "Performance Analysis of Missing Values Imputation Methods ", in Annual Innovative Engineering Research Conference (AIERC 2018), University of Bradford.

- “Performance Analysis of Feature Selection Methods for Classification of Healthcare Datasets ", in in Annual Innovative Engineering Research Conference (AIERC 2018), University of Bradford.

3. Posters

- O. Rado ,“Machine Learning for Sentiment analysis in Health-related Tweets” in Annual Innovative Engineering Research Conference (AIERC 2018), University of Bradford.

- O. Rado ,“Performance Analysis of Missing Values Imputation Methods” in Annual Innovative Engineering Research Conference (AIERC 2018), University of Bradford.

- O. Rado ,“Ensemble of Multiple Classification Algorithms” in Annual Innovative Engineering Research Conference (AIERC 2018), University of Bradford.

- O. Rado ,(under review) titled “The applicability domain of the classification model” in Annual Innovative Engineering Research Conference (AIERC 2019), University of Bradford, 31.08.2019.

- O.Rado, " On Selection of Optimal Classifiers " , Annual Innovative Engineering Research Conference (AIERC 2019), University of Bradford, 15.07.2019.

Acknowledgements

First Thanks to Allah Almighty, for blessings on me and always listening to my prayer.

A special thank you goes to my first supervisor, Prof. Daniel Neagu, for all his support and suggestions.

I want to express my special thanks to Abolgasim Bolomi, my husband, for taking extra care of our children, especially our son Abdulatiff.

I am very grateful to the Ministry of Higher Education in Libya for providing me with the necessary funding to pursue my graduate studies and the Libyan Cultural Affairs for their support and guidance.

Thank you all

Dedication

I dedicate this work to

My brothers, my sisters, my husband who helped me in achieving this PhD, and my children (Fatma, Abdulatiff, and Sohaib), without their support and understanding, this work would never have seen the light of day.

Contents

List of Figures

List of Tables

Glossary

AD	Applicability domain
ADOC	Applicability domain of classifier
AI	Artificial Intelligence
ANN	Artificial Neural Networks
AUC	Area under curve
CAD	Computer-aided diagnosis
CART	Classification and regression trees
CFS	Correlation-based feature selection
dk-NN	k-nearest neighbors' density
DT	Decision tree
FS	Feature selection
KNN	k-Nearest neighbor
ML	Machine learning
MOO	multi-objective optimization
MSE	Mean square error
NB	Naïve Bayesian
PCA	Principal component analysis
QSAR	Quantitative structure activity relationship
RF	Random forest
RFE	Recursive feature elimination
SVM	Support Vector Machine
SMOTE	Synthetic Minority Over-sampling Technique
UCI	Machine Learning Repository
VIMP	Variables importance
μ	mean value of the distribution
σ	standard deviation of the distribution

1 Introduction

The main aim of this study is to investigate the connection between the applicability domain (AD) and the classification model performance. We are examining the usefulness of assessing AD for the classification model, i.e. reliability, reuse, robustness of classifiers. Section 1.1 presents the motivation behind this work; Section 1.2 outlines problem description, Section 1.3 explains the thesis aims, Section 1.4 provides the research questions related to the objectives; Section 1.5 presents the contribution of this study, Data and methodology are listed in section 1.6, Finally, section 1.7 describes the structure of the thesis.

.

1.1 Motivation

Models are often used to support decisions makers in various business sectors. Therefore, it is crucial to ensure model quality to gain useful outcomes. The quality of machine learning (ML) models can be tested based on data quality or algorithm quality. The most excluded aspects of building a machine learning model are assessing the AD of the model, as well as the quality of the data used to construct the model. Feature engineering techniques are used to improve the quality of the model, such as using feature selection methods to remove redundant features that impact the performance of the ML model. Any model of the ML needs not only performance with excellent accuracy but also the reliability of new predictions. Setting data space boundaries where the model has reliable and defined performance is necessary. These boundaries might be the applicability domain (AD) and define the extent to which the ML model (reliably) tolerates new dataset.

The trained model is considered as the core of ML models. However, without guarantees to ensure the quality of these models, the outcomes they give will be suboptimal. Models can take on various structures relying upon machine learning algorithm utilised, but they are data structures containing the parameters learned during the training stage of the algorithm. For instance, a trained decision tree model contains all the splits and values at each node, while a trained k nearest neighbours (KNN) classification model demonstrates the whole training dataset. Model quality is not only crucial during the underlying training and deployment stage. Regular maintenance will guarantee that the model

does not misrepresent after some time. The pipeline of the classification process is graphically given in Figure 1. The classification process is usually divided into a series of steps.

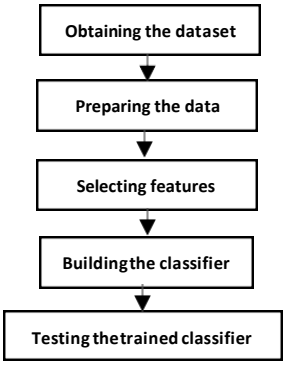


Figure 1: Classification process

There is a vast amount of data generated from daily business/organisations processes; this data is essential to most domains. However, to obtain the actual value of this data, we need data modelling tools for constructing models based on this data. Currently, devices and modelling tools have bee n developed to become more available. Therefore, more users can produce more data without much effort. The topic of big data has gained more interest throughout the last years. Many of these models are useful but they are required time and storage; thus, they may be reused and recycled. Practically, big data is not only a challenge, but the models as well became another dimension in significant data challenges. This model will be assessed for explaining its outcomes, modifying or combining with existing knowledge.

For reused models, the information should come about their applicability because of there is need to know how can reuse them. This research addresses this current problem in context reusing of the model. For example, an expert in a domain receives a question about his expertise in a slightly different way. The outline of this study are big data challenges related to re-usage, recycle characterising and evaluation of machine learning models with consideration of their applicability domain. Applicability domain is a topic used in different fields where required such as predictive toxicology. However, this topic is not extended yet in any area.

Due to these reasons, more research is required to develop efficient approaches for evaluating classification systems and how to incorporate these models in the applicability domain approach to make them useful. The motivation behind this research work is to develop a framework that can help in the evaluation of classification models. This system will also help address in addressing the problem of providing incorrect information such as incorrect diagnoses, as it aimed at assessing the applicability domain of the trained classifier. In this study work, the above-mentioned issues are investigated. The

method is proposed to evaluate the applicability domain for the model of classification. The research is split into sub-problems in this strategy, and these sub-problems are conducted in various attempts:

- Assessing applicability domain for classification model.
- The robustness of the classification model based on the applicability domain approach.
- The selection of a model based on Pareto optimality.

1.2 Problem description

In the data analytic process, there are models created from this data, and there is information about these models. We want to test and evaluate the best of their abilities to find out how much they are useful and how this domain increases in term of time and computation. The research work studies concentrate on offering user-friendly model construction environments [1] Figure 1 shows the pipeline of the classification process. Due to a growing quantity of experimental data and models trained on this collected data, it is not simple to decide which model to use. The absence of techniques for analysing and selecting models can discourage users. Such models have been created for a specific dataset, which may not be able to assess a new dataset. Currently, there are many models, which involve considerable intelligence and processing time. Processing time requires energy, resources. Moreover, making new models maybe not an easy process. There are valuable and available resources of models and data. For example, for models of machine learning techniques such as decision trees used in a variety of fields. Can we reuse them? Our work is an attempt to investigate how we can do reuse them with a specific focus on where are they useful and successful by assessing the AD for these models.

1.3 Thesis aims

The main aim of this work is to explore the evaluation process of machine learning models particularly classifiers with a focus on their applicability domain for health care data. Classification techniques use knowledge discovery process to classify data further, an approach on how to evaluate classification models is proposed. We concern the robustness of classifiers, reliability of classifiers and the coverage issue of the classifiers. The primary goal can be addressed as the following objectives:

- A literature review related to applicability domain and machine learning techniques is performed, and the challenges and gaps are identified.

- Explore the current state-of-the-art techniques for classification model evaluation to identify the gaps in this study domain.
- To propose an algorithm to create a framework for improving the classification model's performance accuracy. Advanced AI algorithms and related methodologies will be used to define the best approach to the suggested framework.
- We highlight the benefits of classification model reuse. Since the quantity of experimental data and the number of classification modes are increasing every day, the development of automated techniques for mining models in repositories is essential. The most challenging task is to locate a model from models' collection for a new dataset.
- Explore and identify the variables that can be selected from the data that could influence the model's AD.
- Implementing the results of the proposed algorithm with multi-objective optimization (MOO) problem to select the highest outcomes.

1.4 Research questions

This research work aims to investigate the usefulness of AD of the classification model to address the challenge of assessing the applicability domain of the classification model by designing a framework for healthcare data. Using Pareto optimality for model selection approach is applying by incorporating the proposed framework. The assessment of whether a classification model applies to a given new dataset is addressed. Assessing whether a given classification model is applicable to a given test set can be broken down into the following two questions. This research particularly investigates the following research questions:

Q1: Can the Applicability Domain be defined such as the Machine Learning classification model will tolerate a new extended data subset reliably? What will be then the effect of the Feature Selection method choice on the assessment of the Applicability Domain?

Q2: How can the robustness of the Machine Learning model be evaluated by considering the Applicability Domain concept in the evaluation of the classification model?

The importance of having an explicit definition of the AD of the model becomes apparent when addressing question 1. Unfortunately, in practice, there is often limited information concerning the model applicability domain. Therefore, the AD technique is of less use for classification models.

1.5 Contribution

The research contributions of this work are as follows:

- Investigation of the applicability domain of classification models.
- Investigate the effect of FS methods in defining the AD of the classifier: The recursive elimination feature (REF) method is applied to the data considered in this research work to investigate the effect of FS methods in defining the AD of the classifier.
- A novel approach is proposed for estimating the robustness of the classification model based on the applicability domain approach. An algorithm inspired by the methods-based distance of assessing the AD is proposed.
- Using the concept of Pareto optimality: Pareto optimality approach is applied to select the best classifier performance from collection of models.
- Extensive literature review: An extensive literature review of the evaluating of the classification models is performed and critically analysed. Literature review related to the applicability domain is performed, and the challenges are identified.

1.6 Data and methodology

This research mainly was motivated by the need for model reuse in the classification problem, and some attempts were presented in the area for healthcare data. The applicability domain approach is used to estimate the applicability domain of classification model in order to evaluate the coverage of the classifiers. This thesis deals with methodological issues that arise in model evaluating and reusing the ML model. The current methodologies were used to accomplish the study objectives:

- Optimal Pareto strategy [2] is used to address the multi-objective model issue for selecting a model.
- Decision trees algorithm [1] and random forests algorithm [3] are utilized as the basis to develop the proposed model.
- principles of the applicability domain approach [4].

Three different attempts will be discussed in this thesis.

The first attempt, investigation of applicability domain of machine learning model for identifying the robustness of the classification model.

In the second attempt, investigation of applicability domain of machine learning model by using density neighbourhood approach.

Finally, classifier automatically selected using Pareto points approach.

1.7 Thesis structure

This study is divided into seven chapters and it is structrued as follows;

The current chapter, **Chapter 1** introduces the work, its context and motivation, and presented the research question with aims and objectives.

Chapter 2 presents an overview of the relevant theoretical foundations covering many aspects of machine learning techniques and, classification algorithms, and applicability domain. The first section includes a definition of machine learning, then classification algorithms as well as ensemble classifier. It is followed by brief the evaluation of the classification model. Next, Convex hull is defined. The section is closed by the applicability domain approach.

Chapter 3 describes the dataset, tools and technologies as well as the methodology of this research work. In this chapter, we describe the dataset, the techniques and methods used, and the different techniques used to implement the proposed approaches. Section 3.2 gives the data set used to train, test and validate the proposed approaches. This chapter also explains the important of the datasets. Finally, description of the research methodology is included.

Chapter 4 presents the results of the approach of assessing the applicability domain (AD) of classifier. AD defines the extent to which a quantitative structure-activity relationship (QSAR) models can tolerate new compounds reliably [5]. This section presented the AD of a classification model (ADOC) proposed. The approach processes of the applicability domain of the classification model are described in detail.

Chapter 5 presents the results of the approach of the robustness of the classification model based on applicability domain approach. This approach was performed in three different stages that efficiently used the features of the AD concept. The proposed method depends on some procedures. (1) Measuring the distances to identify the close points. (2) Using synthetic data to test the robustness of the model. (3) Defining a threshold for each test data. (4) Optimising the threshold parameter r. The distance matrix and the model's response domain are considered to reflect upon the reliability of results derived in its descriptor space.

Chapter 6 provides the results of classifier automatically selected using pareto points approach. This section aimed to apply the Pareto optimality approach for optimising the accuracy, the data rate involved and the threshold of a classifier. This approach considers not only the classification accuracy but also the potential trade-offs between classification objectives.

Chapter 7 summaries the contributions drawn from the work presented in preceding chapters and offers an outlook for possible future research on this topic. Limitations of the research work are also presented in this chapter.

2 Literature review

2.1 Introduction

This chapter presents an overview of the relevant theoretical foundations covering many aspects of machine learning techniques, algorithm types of machine learning, and machine learning applications. Section 2.2 includes a general overview of ML, machine learning types, and prepare data before doing the analysis; Feature selection methods is presented in section 2.2.3. Classification is provided in section 2.3; Ensemble learning for classification model and performance measurements are offered as well in this section. It is followed by a brief description of the applicability domain in section 2.4. The summary closes the chapter.

2.2 Overview of machine learning

Machine learning is a specific application of data science and a subfield of computer science. The ML includes creating and developing algorithms that can learn from data. Due to the acquired popularity of machine learning topic in the last few years, experts in many domains produces an extraordinary expansion of research in machine learning. Since ML algorithms require learning information, the discipline must be connected to the database discipline. Figure 2 illustrates several new areas that have expanded [6] , and some previously established areas have gained new activities such as pattern recognition [6][7] .

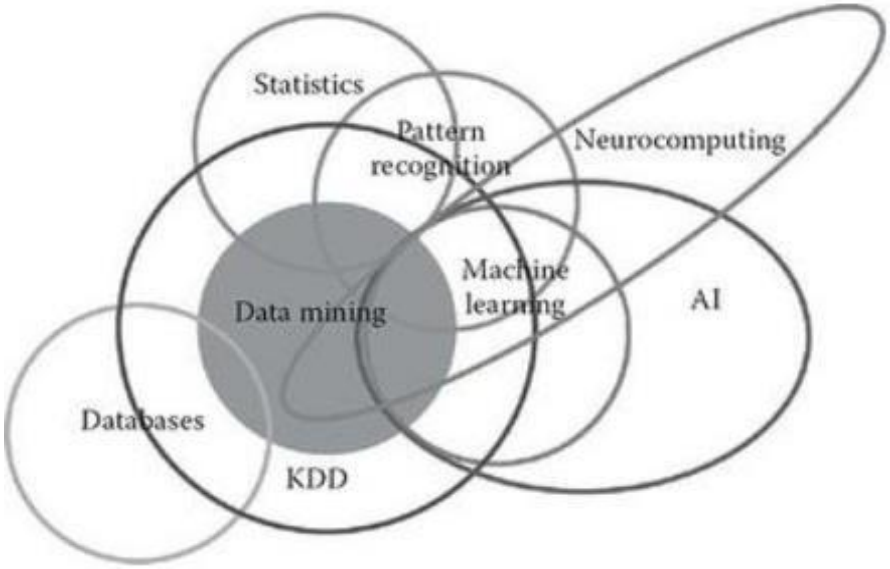


Figure 2: Different disciplines of knowledge and the disciplineof machine learning [7]

Machine learning is about building learning models by using the right features for solving the right tasks. Machine learning algorithms have been used in many various problem's domains. Many publications offer a good starting point to be acquainted in machine learning applications [8][9][10][11][12][13]. The ML techniques have been studied in many different contexts, such as data mining, decision-making, and sensory signal recognition. The types of machine learning algorithms can be categorised based on the learning method and based on the relationship between the learner and the environment. Data mining utilises a combination of statistical rules and rules from the ML area [14]. Data mining explores and analyses dataset to discover meaningful patterns. Figure 3 illustrates the data mining task considering predictive and descriptive outcomes A predictive model creates a prediction about data values using known results of the past data. Whereas, predictive duties include classification, regression, time series analysis and forecasting. A descriptive task defines the relationship in data and explores the data properties. This type of task provides clustering, summarization, association rules, and sequence discovery [15][16]. Developing ML algorithms process can be decomposed into stages as collect the data, prepare the data train, the algorithm, test the algorithm, and apply the validated model [16][17].

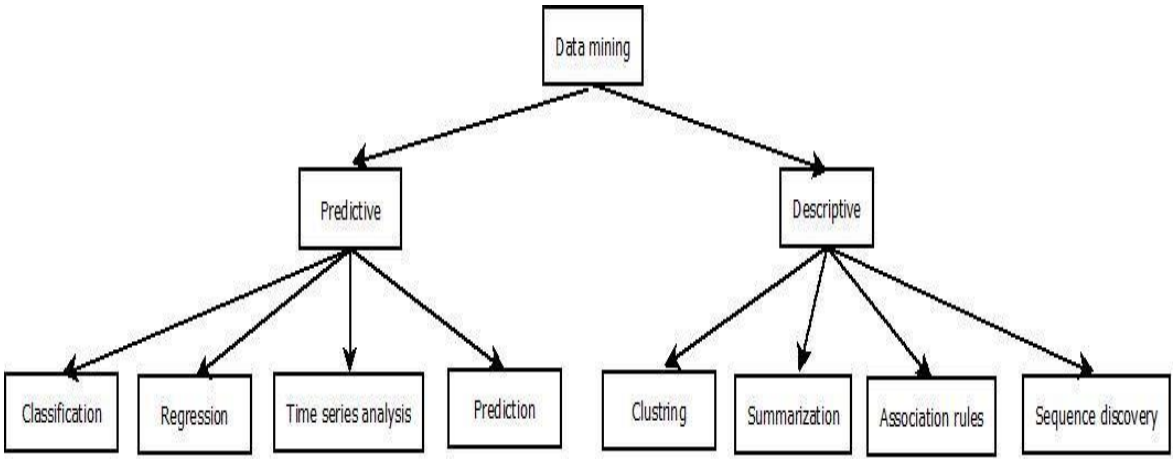


Figure 3: Data mining models and tasks [2]

Figure 4 shows a fundamental process of machine learning project [17]. Acquired data from many sources, it might be data from organisations or open data from the internet. The obtained data needs to be tested for the quality before the analysing. These processes can occur in the prepare phase.

2.2.1 Machine learning types

The field of machine-learning includes supervised, semi-supervised, and unsupervised learners [11][17]. Precisely, training data is available to the learner before building the model, and the testing data are used to evaluate the algorithm [18][19][20]. Supervised learning is about learning a target function from input and output training examples [11]. Unsupervised learning tries to learn input patterns for which there are no output values.

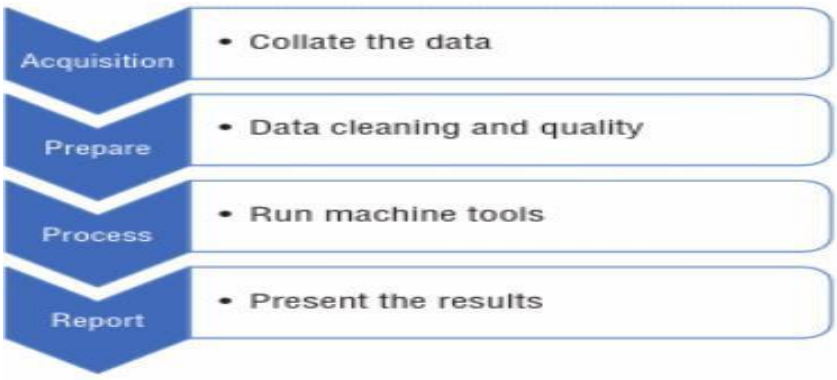


Figure 4: Machine learning process [21]

The field of machine learning includes:

2.2.1.1 Supervised learning

This chapter describes the main points and essential results of supervised learning briefly. For a detailed discussion of the subject, the reader is referred to a study in [21]. We are interested in learning a relationship between the input elements (an input space X) and an output value (an output space Y) values. Mapping function $f: X \rightarrow Y$, the output space Y is discrete as classification. Alternatively, the output space is continuous as regression. Learning depends on a finite training set D of examples is:

$$D = \{(x_i, y_i): x_i \in X, y_i = f(x_i) \in Y, i = 1, \dots n\},$$

Supervised learning refers to the mechanism that infers the underlying relationship between input data and a known class label. An example of labelled data is medical histories that are labelled with the occurrence or absence of a disease [22]. In these cases, the output of the model would be a diagnostic prediction. Learning tasks use the labelled training dataset to combine the model function that generalises the relationship between the input feature vectors and the outputs [12]. A trained function model based on a supervised algorithm can predict the class labels for unobserved data instances such as classification and regression [23]. Generally, learning algorithms aim to minimise the error for a given set of inputs. However, the model may encounter the problem of overfitting, which typically represents unsatisfactory generalisation and erroneous classification. Overfitting results in

high classification accuracy. In the context of the classification task, the function that maps an input x to an output y is called a classifier. In statistics, y is known as a class variable (discrete outputs such as “yes”, or “no”), and x is a real vector. Regression is a learning task in which a target function f maps each Independent attribute x into a continuous output y. Both classification and regression suppose that a training set of examples with real classes or feature values is available [24]. Some applications of regression are (a) demand analysis in the business, including the predict of the stock market index based on economic factors [25], (b) predict the likelihood of readmission of a patient, and healthcare cost prediction in the healthcare domain [16][26].

2.2.1.2 Semi-supervised learning

Semi-supervised learning uses a combination of large unlabelled datasets and a small number of labelled to generate a classifier [27]. Its methodology operates between the guidelines of supervised learning and unsupervised learning for producing good improvement in the performance of learning. In semi-supervised learning, one possible strategy is to use a small labelled training set to construct a model, which is then refined using unlabelled information [28]. For example, we could use the initial model to predict unlabelled data, and then using the most confident predictions as new training data. After that, we could retrain the model on this expanded training set. Recently, semi-supervised learning has been applied in diverse applications, including web data, stock data, images, and biological data [29][30]. This methodology of learning can deliver the value of practice in human learning areas such as speech [31] and vision [32]. These areas involve a small amount of direct instruction and a large amount of unlabelled experience.

2.2.1.3 Unsupervised learning

The unsupervised learning approach discovers hidden structures in the dataset that have no label. This dataset does not contain a class label in contrast to supervised learning, which provides labelled input data. Clustering is the most common approach to unsupervised learning [33] Clustering technique works based on assessing the similarity between examples and placing similar instances in the same group and diverse instances in different groups. The popular algorithm of unsupervised learning is K-mean. It is an approach for representative-based clustering [14][16] Algorithms may employ different measurements, such as statistical measurements and quantisation error. In unsupervised learning, there is no label assigned to the data. Thus, there is no straightforward way to evaluate the accuracy of the outcomes produced by the algorithm. Most important applications of unsupervised learning

are finding association rules. These are important in market analysis, banking security and consists of an essential part of pattern recognition, which is critical to understand advanced AI [16].

2.2.2 Preparing data for machine learning

One of the essential steps of using a classifier to solve real-world problems is to collect and prepare data into a form acceptable to the ML algorithms. The first step in preparing the data is to understand the content of the data, including data meaning and the way the data collected [15] For example, the data type of the attributes is necessary to be understood because the variables of the data set play a role in building machine learning models. The dimension of the data, class distribution and data correlation are essential as well. The independent and dependent variables are used in making the model of machine learning. However, other variables may not be used in building the model but can be used for explanations [34] There are some interactive techniques can be useful to explore the data to better understand its features, such as summary tables, and data visualisation. The steps used to prepare the data to apply includes:

2.2.2.1 Data cleaning

Cleaning the data usually takes a considerable amount of time. Cleaning operations include defining errors and missing values. Machine learning algorithms cannot proceed before solving these problems [24]. Data preparing step refers to treat data set to deliver a predetermined purpose effectively. The quality of input data of a machine learning system manages its success or failure [20] For instance, while training a supervised learning model, feeding an algorithm by training set that includes the majority class and the other is a minority class will not result in a balanced and generalizable model. The resulting system might be great at recognising majority classes, but it will likely be unable to detect the second class. Machine learning algorithms rely on datasets fed into algorithms to execute the learning task. Data is considered Low-quality data based on the degree of fit or meet the underlying expectations based on the context of the Data [20] The data quality metrics allow marking errors resulting from missing or incomplete entries within a set of data. Some causes of data quality problems in machine learning are:

Completeness

Completeness means there are no missing, or incomplete entries of any data elements. Incomplete data is indicated as a lack of quality. Therefore, serious consequences can lead to invalid results in the data mining application when the used datasets are containing rows with missing values [35]. Missing

data is one of the most popular issues in the machine learning field. This issue can be caused by errors in the data collection process or by design [6]. For example, in the dataset gathered by surveys, some people do not answer some optional questions. Thus, the data will contain null values, which cause a problem for the analysis process [14]. Some algorithms use default values in the input data, which can affect the results. However, others will remove such sample even if it contains valid values in most columns. Moreover, excluding the column does not cause a high decrease in performance [35]. Generally, there are four methods for dealing with missing values in the dataset:

- Remove rows with any missing values.
- Exclude columns that contain missing values.
- Replace missing values by mean, median or mode.
- Impute the missing values.

A part of the work presented in this section was published in [36].

Noise

The noise in the data science context is data having inaccurate data (invalid) points [37]. These values are often corrected before running the data mining process because they obscure the actual values of the dataset, such as in image classification problem and errors of the human labellers [38].

Outliers

There are often some data entries that have different characteristics from most of the other objects of the data set [39]. Thus, these values do not fit into the derived model nicely, and the model may behave unwell for the entire data [40].

Relevance

The data should meet the straightforward needs for which they are gathered and for further different purposes. Some features might be irrelevant to the task being developed [40].

Bias in dataset or class imbalance

Real-life datasets might not be well balanced, and thus using unbalanced datasets can result in bias that affects the outcomes [41]. For example, a given diabetes dataset might not vary enough to cover all different types of diabetes that a system might be expected to classify. The dataset depends on what was collected or the source of the data. Typically, challenging to gather samples from the whole universe of data whose behaviour the algorithm should model. Datasets are made by drawing from

sources that produce examples belonging to the world of data. For instance, a given dataset for the disease is X samples for the one month. Utilising this dataset makes a classifier performs very well for X samples. However, there is no guarantee that the classifier will keep on performing great as time progresses or if brought to another dataset. Particularly, Y samples of a similar illness additionally have a place in the universe. Y might have different properties, which are not exhibited in samples X. In this case, it is unlikely that the classifier will produce good results on Y samples. Class imbalance is resulting from data bias in which the number of examples of one class is decidedly smaller than the number of samples of another class label of data [41]. The classifier may cause selection bias because of temporal impacts that added to the choice of the specific dataset used to learn the classifier [42]. Selection bias is a regular type of bias that can be brought about by imperfect data collection flows. A different kind of bias is called observer bias. It caused by errors human-designed processes which cause to gather incomplete data with incorrect labels assigned to samples, affecting the accuracy of the system [42].

2.2.2.2 Transformation

In some situations, it is vital to make new variables from existing variables of the data to enhance the performance of the ML model. For example, calculating the average of a series of numbers, or from date of birth, we can know the age. The principal component analysis (PCA) [42] reviews the connection between variables for extracting a small number of factors representing the data variance [42]. Some attributes need to be scaled in the range of [0, 1].

2.2.3 Feature selection methods

Feature selection offers an efficient way to remove irrelevant and redundant data. Applying FS can decrease computing time, enhance learning accuracy, and promote a better understanding of data or the learning model. This part discuss assessment steps that are frequently used for selecting features [43]. Feature selection methods have been applied to various datasets in several domains, including healthcare. In recent years, there has been an increasing interest in analysing healthcare datasets, which contain hidden information which needs to be extracted for the right decision. According to [43][44], many potential benefits can be obtained with the use of feature selection, such as:

- Reducing the number of irrelevant features and reducing the measurement cost.
- Reducing redundant features and leads to an increase in accuracy and efficiency of the model performance.

The most used feature selection methods include:

2.2.3.1 Correlation-based feature selection (CFS)

Feature selection for classification tasks in machine learning can also be accomplished based on the correlation between features, and such a feature selection procedure can be beneficial to machine learning algorithms [45]. CFS method generates ranks of the feature based on the correlation with the other features. All the features which give less correlation with the rest of the features will be selected and exclude the features that have a high correlation from the data [46] For removing the highly correlated attributes, `findCorrelation()` function from the caret package can be used.

2.2.3.2 Variables importance (VImp)

The variable importance can be quantified by using the score of the importance of given attributes. The use of the mean of misclassification rates for classification or mean square error (MSE) for regression [44].

2.2.3.3 Recursive feature elimination (RFE)

Recursive feature elimination (RFE) [44] is a feature selection method that is used to remove weakest features. RFE seeks to improve generalisation performance by ranking the features and recursively removing the least essential features whose deletion will have the least effect on training errors [47]. In general, FS refers to the method of acquiring a subset from an original feature set by selecting the appropriate attributes of the dataset according to specific selection criteria. It plays a part in data processing, removing redundant and meaningless features. Feature selection method can enhance the accuracy of learning, decrease learning time and simplify learning outcomes.

2.2.4 The use of machine learning

This section describes how machine learning is being used in real-world applications. Machine learning is widely used in a variety of domains. The proposed approach builds on the training dataset and then classify test dataset based on the learned knowledge. Many areas of mathematics and computer science, including machine learning, allow data searcher to offer valuable services to almost any field. In the healthcare field, there is a large amount of crucial clinical data that might be useful for data research. The datasets used to validate this work are from the healthcare domain. Therefore, the use of machine learning in the healthcare domain will described.

2.2.4.1 Machine learning in healthcare

The ML techniques in the healthcare field provide healthcare professionals with better information for making a better decision. Machine learning (ML) technique are powerful and flexible tools for analysing and predicting results from clinical or biological data. The ML model has the potential for improving healthcare in many ways efficiently. Healthcare datasets are optimal targets fordata mining methods. Several data mining techniques have been applied to healthcare and medical data for predicting many diseases. Some algorithms that are used to predict prognosis empower healthcare officials to allocate resources optimally and physicians to provide better treatment opportunity for patients. The main areas of possible applications of machine learning in healthcare domain are:

1. Medical diagnosis

In the healthcare field, practitioners can use diagnostic models for recommending appropriate testing and treatment. The diagnostic models can help to decrease the burden on physicians, increase patient access to care, reduce costs, and save resources. However, despite the research advances of ML techniques in the healthcare domain, its role in the clinic is still limited [48]. The accurate diagnosis of diseases at the early stage plays a significant role in the care of patients. The ML tools can detect the importance of the features from big and complex datasets. Diversity of the ML techniques including support vector machine (SVM), decision trees [18], deep learning (DP) [49] and neural network (ANN) [50] applied in diseases [51] diagnosis such as Alzheimer's disease heart disease [52] and cancer [53] for building models of classification and prediction. Pattern recognition algorithms used in computer-assisted diagnosis can help physicians interpret medical images in a comparatively brief period. Medical images from various medical exams such as X-rays, MRI and ultrasound are the information sources that describe the situation of a patient [6]. Application for computer-aided diagnosis (CAD) is to identify and classify breast lesions in ultrasound pictures [28].

2. Drug discovery

Drug discovery involves a broad range of scientific disciplines, particularly in the areas of biology and chemistry. It is the process to identify potential medicines that influence diseases. The ML and artificial intelligence (AI), including deep learning, are powerful methods for understanding the conditions that affect molecules because they can deal effectively with high dimensional databases. The ML approaches that are commonly used in drug discovery are SVM [8], DT [54], k-NN [55], naïve Bayesian methods [28] and ANNs [42].

3. Treatments

Disease treatment is a process for Identifying what disease that a patient suffers from to determine appropriate treatment can be given to that patient. The ML methods in many applications are used in disease treatment to identify what type of disease, prevent and side-effects [56]. Machine learning applications of robotics are used in surgery [13] [56][57]. The healthcare industry can be considered as a place with rich data due to generating massive amounts of data from administrative reports and electronic medical records. Healthcare field covers some detailed processes such as prevention of disease, the diagnosis, treatment, and injury. Solanki et al. presented a study of the analysis of some application of data mining techniques in healthcare [57]. Their study provided a comparative accuracy analysis of various data mining techniques in the healthcare sector. Some of the data mining techniques are considered including decision tree classification, support vector machine classification, linear regression, hierarchical clustering. Researchers use those techniques as they provide high accuracy and efficiency.

Some studies investigate approaches for exploring the data mining and healthcare industry fields. These fields have arisen some of the various reliable systems of early detection and different healthcare-related from the clinical and diagnosis data. Jothi et al. have investigated the different paper associated with this field in terms of method, algorithms and results [58]. The ML and big data topics have gained much attention from researchers in healthcare [59]. Manogaran and Lopez presented a survey of big data architectures and machine learning algorithms in healthcare [59]. Their study includes an overview of the state-of-the-art machine learning algorithms to process big data in different domains, including healthcare [60][61]. Following are the different classification algorithms applied in healthcare:

One of the most common classifiers is decision Tree (DT) is considered [22]. Decision tree algorithm is used to analyse the clinical data. Some studies have explored the decision tree algorithm in their work [57][60]. All the three works have used the decision tree to the data set to increase the prognostic performance depending on the accuracy. The used data in these researches are balanced data set. The k-nearest neighbour is a distance-based classifier method. Studies by Bagui et al. [58], Armañanzas et al. [62], and Jen et al. [63] have used the k-nearest neighbour in their respective predictive models. k-NN performs well for a large and homogeneous dataset. However, it has no explicit model, so all the calculations have to be repeated to classify a new case. Support vector machine (SVM) method is commonly used in medical diagnosis.

Studies by Suet al. [64] and, Zheng et al. [65] have used the SVM technique for medical diagnoses. According to the results obtained from these comparative studies, SVM provided the best performance because it maps the features to high-dimensional space. Moreover, SVM can handle

classification tasks with excellent generalisation performance. Gupta et al. presented an approach for recognisable proof and forecast of MicroRNAs (miRNA) in infections using artificial neural networks (ANN) [66]. MicroRNAs (miRNA) are a class of non-coding RNA They used structural characteristics of pre-miRNA to prepare the ANN for identifiable evidence of miRNA in new viral genomes. The results demonstrate that this system might be used for distinguishing novel miRNA as a part of other viral genomes with respectable triumph.

Another study also evaluated the performance of five different classification methods, including C5.0, Rpart, k-nearest neighbour (KNN), SVM, and random forest (RF). Three different feature selection methods are applied, including the correlation-based feature selection method, variables importance selection method, and recursive feature elimination selection method. Seven relevant numerical and mixed healthcare datasets are considered. Ten-fold cross-validation is used to evaluate the classification performance. The experiments showed that there is a variation of the effect of feature selection methods on the performance of classification techniques [67]. According to Miotto et al, using deep learning technologies to advance the health care domain could be the vehicle for translating important biomedical information into improved human health. However, there are constraints for enhanced growth of techniques and apps, mainly for domain specialists in terms of ease of understanding [68][69].

The ML model of classification can be useful to diagnostic systems for the disease. For help in the diagnosis process, software applications ("apps") were created. In the Google Play and Apple App stores, these apps are accessible [70][71]. Further, these systems provide only the facility for the diagnosis of certain illnesses. However, they also increase the danger of incorrect data being presented. Further study on the use of these applications, the consequences for medical practice is needed [71], especially in the applicability domain of these applications.

Although there are many medical accomplishments, some illnesses continue to plague humanity [72]. Diabetes, cardiovascular diseases, cancer, chronic respiratory diseases and mental disorders represent approximately 77% of the disease burden and 86% of fatalities in the European region [72]. Coronary Heart Disease (CHD) is one of the leading causes of death globally [73]. Alzheimer's disease is affecting about 60 per cent of demented people [72]. Early diagnosis of this disease will assist patients in leading a quality life for the rest of their lives. Diagnosing the existence and severity of any illness correctly usually includes using a costly operation. One feasible alternative is to use computational methods to predict any cases of the disease to provide an original estimate of its probability. According to Marmot [74], policymakers in every industry should be concerned about health status, not just those engaged in health policy. Using healthcare data can enhance public health services and help define disease

patterns that may lead to more efficient prevention procedures. It is commonly acknowledged that routinely collected health care data can be used to enhance the health of communities. Data analysis by machine learning provides significant benefits in evaluating large quantities of complex data on health care [75]. The accuracy of the classifier depends heavily on the reliability of data in terms of its clinical reality reflection. For example, if an accidentally incorrect medication is included as part of a dataset. Then this dataset utilised to train a classification model to suggest treatments for an illness. The trained model could wrongly recommend this medicine for a condition, resulting in disastrous consequences. Using the skills of robust methods for analysing large quantities of complex health care data can develop health care delivery's effectiveness and cost-effectiveness. Deep learning [76][48] and machine learning [77] are powerful tools for the health care domain. Some of the current work on disease classification is outlined in Table 1.

Table 1: Classification algorithms for diseases

Author	Disease	Resource of Data	year	Classification Technique	tool	Classification Results
Mohan et al.[78]	Heart disease	UCI	2019	Hybrid random forest and a linear model (HRFLM).	R studio rattle	88.7%
ChenWu et al [79]	Fatty liver disease (FLD)	Diabetic Research Institute in Chennai	2019	(RF), (NB), (ANN), logistic regression (LR)	WEKA	87.48, 82.65, 81.85, and 76.96%
Mostafa et al.[80]	Parkinson's disease	UCI	2018	Decision Tree, Naïve Bayes and Neural Network	WEKA	DT, 91.63% NN, 91.01% NB, 89.46%
Sivasakthivel and Shrivakshan [81]	*thyroid data*	UCI	2017	*J8, CART and Random Forest*	WEKA	NM
Bhagya and Sheshadri [72]	*Alzheimer's disease*	*NM*	*2018*	*Naive Bayesian*	*WEKA*	96.69 %

The following findings are concluded based on the literature review undertaken in this thesis on classification algorithms for healthcare data:

- Most of the literature focuses on classification accuracy, and few studies have considered many criteria.
- The size of the data set used small in some research. As few data are used for training and testing; there is no generalisation of the trained models from different sources to new data.

- The classification process must be highly accurate as it is used to assist the specialists in the healthcare field in their findings. However, the accuracy performed based on different factors such as a real database, optimal set of features. Therefore, the performance of the classification should be evaluated appropriately.

2.3 Classification

Classification is the commonly used methods of data mining in many sectors. It is a supervised learning approach, having known class categories. An appropriate machine learning modelling technique can be selected depending on the prior knowledge of the available data type. Some machine learning techniques are hard to interpret, like neural networks, whereas other methods such as decision trees are viewed as progressively straightforward. Decision trees produce a pathway to an expectation that can be followed.

Moreover, they have fast learning and classification abilities. There are different techniques available, but this section focuses on those most used in healthcare data modelling. To better use classification algorithms as tools to solve real-world problems, we need to have a clear understanding of both the issues, the algorithm, and the methodology used.

This section presents background on classification as supervised learning. Section 2.3.1 covers classification algorithms, Section 2.3.2 covers ensemble learning for classifiers, and Section 2.3.3 discusses performance measures. Classifier evaluation is presented in section 2.3.4. Section 2.3.5 includes the classifiers quality metrics. Some studies of robustness in machine learning are presented in section 2.3.6.

2.3.1 Classification algorithms

Generally, there are distinct kinds of learning techniques, each having its own characterises and way for solving some learning issue. Classification algorithms have been used successfully in many domains. Well-known machine learning algorithms are C4.5 [82], SVMs [28], RF [3], AdaBoost [83], K-NN [12], classification and regression trees (CARTs) [19][84], and naïve Bayes [28]. Some ML classification algorithms consider a probabilistic classification approach. A probabilistic classification approach aims to estimate the joint probability density function for each class. In this section, we discuss different algorithms of supervised learning method including, DT and RF, NB, SVM, ANN, and KNN.

2.3.1.1 Decision trees and random forests

Classification trees emulate human reasoning because they are interpretable as a set of rules or as a tree-like flow of information [62]. Decision tree classifier is one of the most common classification techniques. It uses data, which contains feature vectors assigned to a specific class. This approach is used to classify data and to represent the results in a tree structure [6]. This model classifies data in a data set by flowing from the root through a query structure until it reaches the one-class leaf. It is splitting the data set recursively based on the attribute, which divides the data until a specified stop criterion is reached. The domain is divided into regions (subsets) [12]. For a given input dataset d, different split points are assessed for each variable in d. The decision of numeric attribute is of the form $x_1 \leq V$ for V value in the range of x_1. Categorical attribute decision is of the form $x_1 \in V$ for the subset of values in the x_1 domain. The best split point is selected to split the data into subsets, d_{yes} and d_{no}. The points $x \in d$ that satisfy the split decision are in d_{yes}, and d_{no} corresponds to the points that do not satisfy the decision of splitting. The recursive partitioning process can be stopped by using stopping conditions, including the size of the partition d. This condition prevents overfitting the model because the model avoids dealing with a small subset of the data [12][28].

There are some criterions used to split point for a numeric or categorical attribute. A split point which provides the best separation between the class labels was selected. Entropy measures the amount of uncertainty in a system. The entropy of a set of points d is defined as:

$$H(d) = -\sum_{I=1}^{k} P(c_i|d) \log_2 P(c_i|d) \quad (1)$$

Where, $(c_i|d)$ is the probability of class c_i in d, and k in the number of classes. When d split into d_{yes} and d_{no} ,the entropy is given as:

$$H(d_{yes}, d_{no}) = \frac{n_{yes}}{n} H(d_{yes}) + \frac{n_{no}}{n} H(d_{no}) \quad (2)$$

Where, n is the number of points in d, and n_{yes} denotes the number of points in d_{yes} , and n_{no} denotes the number of points in d_{no} . Information gain is used to check the overall entropy. Information gain is given as:

$$Gain(d, d_{yes}, d_{no}) = H(d) - H(d_{yes}, d_{no}) \quad (3)$$

The greater value of information gain indicates the reduction in entropy and thus the better split.

Gini index is another measure to check the purity of split point, it is defined as:

$$G(d) = 1 - \sum_{I=1}^{k} P(c_i|d)^2 \quad (4)$$

The higher Gini index values denote disorder, and the order is denoted by lower values of Gini index.

The weighted Gini index of split point is calculated as:

$$G(d_{yes}, d_{no}) = \frac{n_{yes}}{n} G(d_{yes}) + \frac{n_{no}}{n} G(d_{no}) \quad (5)$$

Where, n is the number of points in d, and n_{yes} denotes the number of points in d_{yes}, and n_{no} denotes the number of points in d_{no}.

The measure of Classification and Regression Trees (CART) is defined as:

$$CART\ (d_{yes}, d_{no}) = 2 \frac{n_{yes}}{n} \frac{n_{no}}{n} \sum_{i=1}^{k} |P(c_i/d_{yes}) - P(c_i/d_{no})| \quad (6)$$

Where, n is the number of points in d, and n_{yes} denotes the number of points in d_{yes}, and n_{no} denotes the number of points in d_{no}.

Decision trees can be visualised as a tree-structured representation form, which is easy to understand and interpret. The model of the decision tree contains internal nodes and terminal nosed (leaves) that assign class labels to regions. Iterative Dichotomiser 3 (ID3) algorithm was the first algorithm concerning decision tree training [85]. The C4.5 and C5.0 algorithms improved upon ID3 by dealing with missing data, performing pruning, dealing with continuous data, achieving splitting and rules [82]. Ross Quinlan developed them in 1986 and 1993 [82][86][87]. Over the years, different algorithms have been developed for DT. The most common algorithms are ID3, C4.5, C5.0, CART, and CHAID [17].

The RF model, as presented by Breiman [3] is a combination of many decision trees classifiers. The RF is a robust supervised machine algorithm used for regression and classification problems. Each tree is grown based on two procedures. The first procedure is to build the bootstrap ensemble model, as we demonstrated in Section 2.3.2. At this stage, a subset of the training dataset is selected independently for all trees in the forest. The rest of the examples are called out-of-bag (OOB) set and are utilised to assess the RF's goodness-of-fit [3][33]. The second procedure is for growing the tree by splitting the local training set at each node based on the value of one attribute from an arbitrarily chosen subset of variables. Due to the lack of pruning; therefore, each tree is grown to the most substantial extent possible. The stages of the bootstrap and growing require a contribution of random input vectors which are independent between trees and identically distributed. Thus, each tree is sampled independently from the ensemble of all tree predictors for a given training dataset [3]. Prediction of new data is performed by aggregating the predictions of the trees. It is generally known that methods

are used to combine outputs of the classifiers such as voting (for classification) and averaging (for regression). Table 2 shows the advantages and disadvantages of model-based trees [87].

Generally, the RF model is an ensemble learning algorithm which can help to improve machine learning results by combining several models. This approach produces better predictive performance compared to a single model [3][88][77]. There are many good reasons to utilise decision trees model. For one example, DT is easy to read. In contrast, the main disadvantage of DT is that DT can create complex model, based on the data included in the training set. With a list of advantages of Decision trees, there is usually a set of limitation sitting in the background, as shown in the Table 2.

Table 2: Advantages and disadvantages of model- based trees

Advantages of decision trees	Disadvantages of decision trees
• High readability. • Fast learning and classifying. • The classifier can handle large set of data. • The classifier can handle different data types [28][42].	• Decision trees can be unstable. • Decision trees can have overfitting. • The classifier replicates parts ofthe trees. • Numeric attributes can lead to extensive branching and generate complex decision trees [18][28][89].

2.3.1.2 Artificial neural networks (ANN)

Neural networks have been utilised in many applications such as pattern recognition [90], forecasting [25], classification [65] and prediction [28][9]. The artificial neural network contains nodes, neurons, and weighted connections between these neurons, as shown in Figure 5. In the learning process of the network, weights are adapted. An activation function defines the output value of each node depending on its input values. The change in weights is represented using Hebb rule as:

$$\Delta w_{ij} = c x_{ij} y_j \quad (7)$$

Where, c is a constant called learning rate, x is the input, and y is the output.

Figure 5 shows a multilayer feed-forward neural network. Each neural network contains the input layer, the output layer and hidden layers. The input layer obtains the data from external sources (attribute values), the output layer generates the output of the network, and hidden layers link the input and the output layer [18][50]. The input value of each node in every layer is calculated by computing the sum of all incoming nodes, then multiplied with the weight of the interconnection

between the nodes [10][18]. Many types of activation functions can be used. The sigmoid function is commonly used to calculate the output value by using all input values. The sigmoid function is defined as follows:

$$f(x) = \frac{1}{1 + e^{-x}} \tag{8}$$

Where, x in the input and f is the output. where ij is the sum of the input nodes of j.

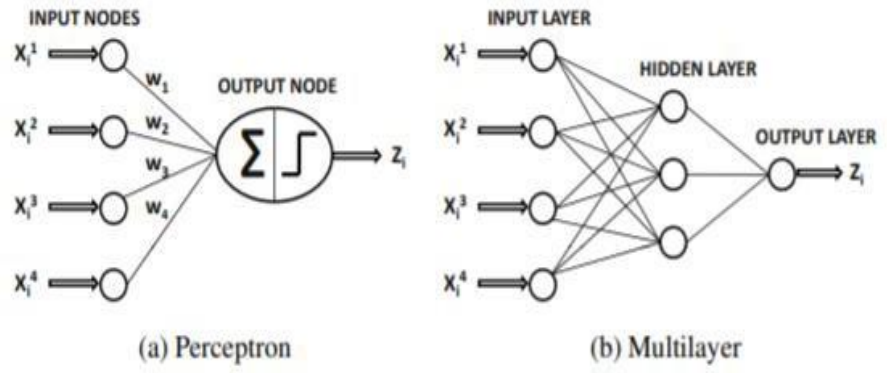


Figure 5: Single and multilayer neural networks [21]

Two main types of neural networks classifier are Feedforward Neural Networks and Recurrent Neural Networks [84]. Backpropagation (backward propagation of errors) is a standard method for training a neural network. Backpropagation algorithm adjusts weights in the neural network based on the error rate obtained in the previous iteration (epoch) [85].

2.3.1.3 Support Vector Machines (SVM)

Support vector machine classifier (SVMs) is a powerful supervised learning technique used for classification and regression. The SVM classifier is among the most accurate methods in all well-known data mining algorithms. The basic SVM algorithm was developed by Vapnik in the mid-1990s as a result of the use of developed concepts of statistical learning theory [19]. The preliminary objective of SVM classification algorithm is to find a hyperplane which can separate the classes (two classes) of given data points with a maximal margin, and for the ability of generalisation. Figure 6 illustrates the hyperplane obtained with SVM, on two-dimensional and two classes (a linear SVM). The dark points represent the support-vector, whereas the hyperplane is corresponding to the classifier. The SVM decision boundaries in the feature space, which separate data points belonging to different classes [77]. Their basic principle of SVM is to construct a maximum margin separating hyperplane or a

function $g(x) = w^T x + b$ in features space [54]. For a given dataset x_i that belong to two classes $\omega 1$, $\omega 2$, the distance from an example to the hyperplane is equal to $\frac{|g(x)|}{\|\omega\|}$.

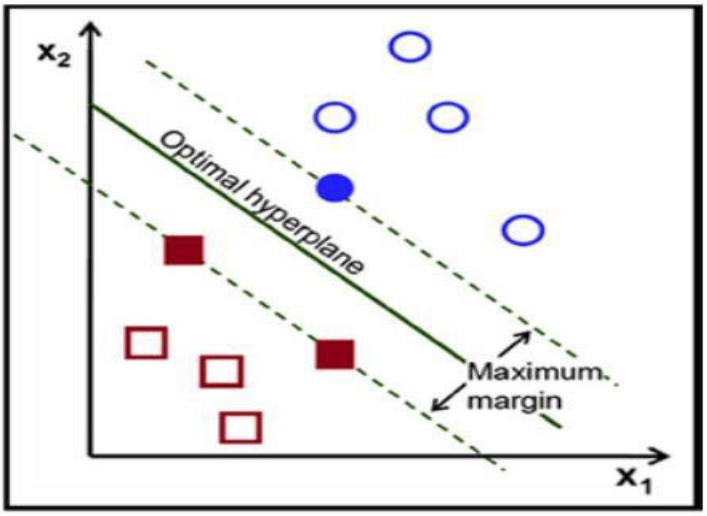


Figure 6: Visualization of support vector machine algorithm finds the hyperplane that maximizes the largest minimumdistance between the support vectors [19]

SVM finds w, b such that the $g(x)$ equal to 1 for the nearest examples belong to class $\omega 1$, and -1 for the closest data points of $\omega 2$. If the problem is not linearly separable in features space, a kernel SVM can be used to transform the data to kernel space (higher-dimensional feature space) [54]. Then learns the optimal linear hyperplane in the feature space. A decision function is identified depending on the linear hyperplane. Kernel function relies on a subset of the training dataset called support vectors [11] classifier establishes.

2.3.1.4 Naïve Bayes

Naïve Bayes approach uses probability theory to find the most likely classification. It assumes that the features are all independent [6][27]. An estimation of the posterior probabilities of the class is determined dependent on feature information. It estimates the posterior probabilities of class C_i, and selects the class with the highest estimated probability:

$$\hat{y} = argmax\{P(c_i/x)\} \quad (9)$$

Where, x is a set of samples, $\hat{y}$ is the predicted class for x, and $P(c_i/x)$ is the posterior probabilities of class c_i .

Naïve Bayes approach can gain knowledge about the state of attributes and their dependencies. The likelihood can be decomposed into a product of probabilities which results in the formula:

$$P(c_i/x) = \frac{P(x/c_i).\,P(c_i)}{P(x)} \quad (10)$$

Where, $P(x/c_i)$ is the conditional probability that x occurs if event c_i is known to be true (the likelihood), and $P(x)$ is the probability of x occurs from any classes [33].

2.3.1.5 k-nearest neighbour (KNN)

K-nearest neighbour (KNN) is one conventional distance-based algorithm for classifying objects based on the outcomes of the closest objects in the training data [91]. The KNN classifier uses the distance (similarity) between the test point and each data points in the training dataset. Next, selecting the K closest points and making a vote of their class labels for determining the label of the test point [92]. However, classes with more frequent outcomes tend to dominate the test object classification. The K closest instances from the training set are considered only. Then, the class of the new point is placed based on most members from this set of K closest instances [93]. The necessary components of the nearest-neighbour classification method [42][7] include the following steps:

a) Take a set of labelled objects with features.

b) Calculate the distance between objects in the training set. A distance metric is a real-valued function d, such that for any data points x, y. $d(x, y) \geq 0, and\ d(x, y) = 0\ if\ and\ only\ if\ x = y$. The most popular distance function is Euclidean distance, which is computed as $d(x, y) = \sqrt{\sum(x_i - y_i)^2}$, where, x_i and y_i denote the attribute values of two points.

c) Consider the nearest neighbours (k).

   K is a user-defined constant, and a test sample with given variables is classified by assigning to the most frequent label among the k training set nearest to that test sample [13].

d) Determine the most frequent classification.

The predicted class for x is: $\hat{y} = argmax\{P(c_i/x)\} = \mathrm{argmax}\{\mathrm{k}_i\}$. Where, $\hat{y}$ the predicted class, and $P(c_i/x)$ is the posterior probabilities of class c_i, and k_i indicates the points number among the K nearest neighbours of x that are labelled with class c_i.

2.3.2 Ensemble learning for classifiers

The ensemble approach combines a set of weak classifiers for improving the performance, especially for the unstable classifiers. The classifier can be an unstable model if a small change in the training set affects the outcomes significantly [6][27]. For example, the decision trees classifier is susceptible to noisy data and tend to have overfitting. Due to constructing a classifier that is robust to noisy data, different classifiers are trained on different data subsets to provide independent outcomes. Next, the results are combined in the way of ensemble learning. The method of selecting the training sets are different. In the training stage, we choose the ensemble size k and the base classifier model for a given data set $X = \{x1, \dots, xN\}$. We make k number of samples from X and train classifiers $C1(x), \dots, Ck(x)$ for all samples. Each sample makes one classifier [26][94].

The decision is taken by voting or averaging. Taking the label assigned by classifier Ci to be a "vote" for the respective class, assign to x the class with the largest number of votes among the classes. Majority voting among the classes $C^k\ (x) = argmax\ \{V_j\ (x) | j = 1, \dots, m\}$. Weighted voting combination over the outcome of the base learners is the way for combining binary classifications can be by having weights $(w1, w2, \dots, wk)$ which deal with ensemble models. Suppose the classes are given as {+1, -1}, classifying the new data points by all classifiers $C1(x), \dots, Ck(x)$ to gain the prediction $\hat{y}$ is expressed as:

$$\hat{y} = sign\ (\sum_{i=1}^{k} wi.\ Ci(x)) \tag{11}$$

Various ways are used to achieve ensemble learning including using different ML algorithms, different parameters (such as trees size or depth), or different training sets. A part of this work presented in this section was published in [94]. The most popular methods are:

2.3.2.1 Bootstrap aggregation (Bagging)

In bagging, different samples of training sets are selected with replacement from the original input training set. Models are trained based on each sample. Each training set is different, with an emphasis on the variance of training instances. The RF classifier is a supervised machine learning algorithm which uses an ensemble of decision trees classifier [3]. The RF select samples randomly by either subset of training instances or subsets of features of each decision point. The advantage of bagging is obtaining low variance due to the averaging effect of majority voting [89][95].

2.3.2.2 Boosting

Boosting is another technique to train the base classifiers on different samples. Moreover, this way raises the performance to classify instances by selecting the samples. The process started by selecting an initial training sample to build a classifier C1 with an error rate. Next, training samples are constructed by choosing the misclassified points with higher probability. The second error rate is obtained with the second classifier C2. To build the third training sample set, the instances that are misclassified by C1 or C2 are selected. The process is repeated for many iterations. Weighted samples or biased samples are employed to obtain different training sets. Finally, a combined classifier is obtained. The advantage of this way has an error rate, which is less than the error rate for a random classifier [96].

Moreover, the classifier C2 may classify some instances where the classifier C1 fail. The idea behind boosting method is to train a new model based on the errors of the previous model and discover the samples that are difficult to classify. The later classifiers focus on these instances better [91] .

2.3.2.3 Adaptive boosting (AdaBoost)

AdaBoost is an example of the boost classification task. AdaBoost algorithm trains N boost models using the weighted trainer. Different machine learning algorithms can be used for this method. It works well on many of the machine learning problems such as speech recognition [31] and face detection problems [32]. Example of AdaBoost in face detection problem is an algorithm called Viola-Jones face detector [97].

The theory behind Boosting Algorithm is to build a classifier on a given dataset, as illustrated in Algorithm1. A weighted classifier Ci is trained on the data x with corresponding classes y, and the weight vector w. The vector w assesses the importance of obtaining data point's right. Moreover, we can know which points are important. Then, all the points weight order to focus on some points in the next learner. Next, we compute predictions of the model $\hat{y}$ Next, compute the weighted error rate for the classifier overall data points to check the points that have poor outcomes. Compute coefficient that is used in weighted updating. It is derived from error as inline 5 in Algorithm1, and then we compute new weights as inline 6 in Algorithm1.

The process of building a classifier and updating weights process is repeated until no more misclassifications are obtained. The final classifier will be obtained by voting from all N classifiers and weighted summation of the outcomes. The result will be above zero for (+1), and below zero for (-1).

$$(sum\ \alpha i * predict\ (Ci, test)) > 0$$

If $y = \hat{y}$ the weight will increase, and if $y \neq \hat{y}$ the weight will decrease. The weight w is normalised to 1.

```
Input dataset (x, y)
      For i=1 to boostNumber do
        Create w
          Ci = train(x, y, w)
          ŷ = predict (Ci, x)
          e = w * (y ≠ ŷ)
          w = exp (−αi * y * ŷ)
          w = w/sum (w)
      End for
End
```

Algorithm 1: AdaBoost process

2.3.3 Performance measures

This section provides different measures to quantify classification algorithms performance. Performance measures that apply to the classification models are presented. Criteria specific to certain classification problem domains are divided into the binary classifier and multiple classifiers. Classification model evaluation is essential for assessing the quality of the classification model. The main objective of evaluating the classification model is obtaining a reliable assessment of the quality of the model results. Generally, two topics related to the evaluation of classifier are performance measure and procedure of the evaluation process [98]. Various performance measures are used to evaluate the efficiency of the classification model by researchers in the literature. Most literature work presents a binary classification. For binary classification accuracy, sensitivity (recall), specificity, positive predictive value (PPV), negative predictive value (NPV), the area under the curve (AUC), ROC curve, Precision, and F1-Score performance measurements are mostly used. These performance measures are given in this section. Binary classification is the most common classification assignment. The input is categorised as one of two non-overlapping classes (C1, C2). Whereas for multi-class classification, the input must be classified into a non-overlapping class of k [99].

The work has been done on different classification tasks are briefly described in the subsections below. Some of the classification models for binary classes, while others consider multi-class classification. Therefore, the classification work has been divided into binary class classification and multi-class classification.

2.3.3.1 Binary classifier

Generally, the model can be defined as the way to represent the data used for the training process based on certain assumptions. After building a machine learning model and obtaining the output in the form of classes or probabilities, the effect of the model is assessed based on several metrics using unseen data. In classification and regression problems, there are many metrics for measuring performance. Different performance metrics can be used to evaluate binary classification models such as Accuracy, Precision and Area under Curve (AUC). One of the best approaches to illustrate the performance of machine learning programs is a confusion matrix. It is a way of assessing the accuracy of models. It visualises the performance of the predicted classification against the actual classification in the form of false-positive (FP), true positive (TP), false negative (FN) and true negative (TN) information. There are two classes for binary classification; thus, the entries of the resulting are 2×2 confusion matrix with four possible cases [18][100][101], as shown in Table 3.

Table 3: Confusionmatrix for binary classification

Data class	Positive	negative
Positive	TP (true positive)	FN (false negative)
negative	FP (false positive)	TN (true negative)

From the confusion matrix shown in Table 3, the accuracy can be computed, as shown below:

Accuracy is a performance measure used to evaluate ML models. Using accuracy is a good indicator in the model evaluation process when the class distribution of the training dataset is well-balanced.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{12}$$

Where TP is true positives, TN is true negatives, FP is false positives and FN is false negatives. In binary classification, the accuracy measure is not an efficient measurement for imbalanced data because the classes in the target variable are a majority of one class. Classification error can be computed by:

$$\text{Misclassification Error} = \frac{1}{N}\sum_{i=1}^{N} I(y \neq y_i) \tag{13}$$

Where $\hat{y}$denote the predicted class of the classifier, y denote the true class, and I is an indicator function that has the value 1 if its argument is true and 0 otherwise. The better classifier the lower misclassification error (Error rate).

Sensitivity (True Positive Rate) is the true positive rate, also referred to as recall. It is the number of instances from the positive class that predicted correctly, divided by the actual number of positive observations:

$$Sensitivity\ (Recall\ ) = \frac{TP}{TP + FN} \tag{14}$$

Specificity (False Positive Rate) is the number of instances from the negative class (second class) that were predicted correctly, divided by the total number of actual negative observations:

$$Specificity\ (TNR) = \frac{TN}{TN + FP} \tag{15}$$

Receiver Operating Characteristics (ROC) is a commonly used metric for evaluating the performance of binary classifiers (e.g. two classes). This curve plots sensitivity on the y-axis and specificity on the x-axis. Area Under Curve (AUC) has values in an area of 1.0, which represents the degree of the accuracy of the model. A point above the diagonal line denotes an accuracy that is better than a random prediction. Conversely, a score below the diagonal indicates that the accuracy is worse than a random prediction. The AUC represents the ability of a model to discriminate between positive and negative classes. An area of 0.5 represents a model as good as random. The random classifier in the ROC plot corresponding to a diagonal line. The better result

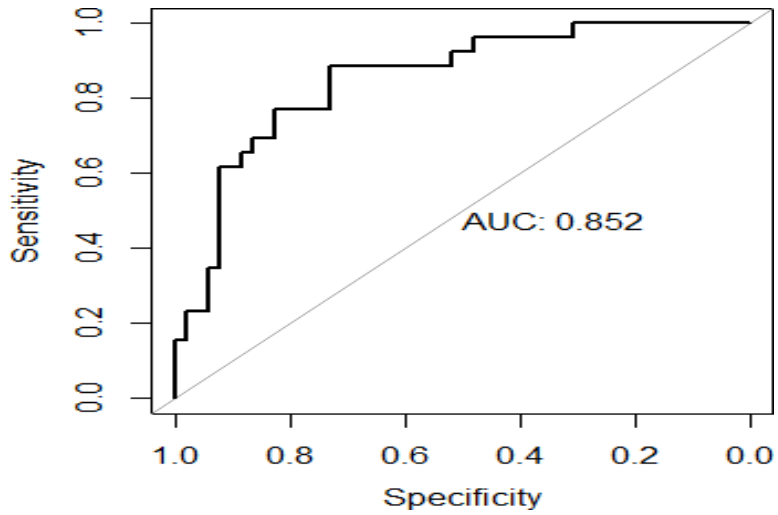


Figure 7: ROC curve for Pima dataset

appears closer to the top-left point in the plot. Figure 7 shows an example of the ROC plot, with the shaded region showing the AUC. The ROC measures the performance of a classifier on Pima

data set, we used the linear discriminant analysis (LDA) [28] to classify Pima dataset. The AUC is 0.85 in this example, which is close to the maximum (top-left corner of the plot). Therefore, the classifier achieved good performance. Plot ROC curve is obtained for the two-class classification model. We used the linear discriminant analysis (LDA) to classify Pima dataset.

Precision can be computed by dividing the number of correctly predicted positive observations by the total number of predicted positive observations:

$$Precision = \frac{TP}{TP + FP} \tag{16}$$

2.3.3.2 Multi-class classification

In this section, some measures of the multiple classification are explained. Multiple classifier evaluation refers to the process of comparing the outcomes produced by the classifier on a given dataset with the actual classes. Machine learning algorithms used measures such as accuracy to quantify performance. For the classification problem, the predictive accuracy of a model can be estimated by the correct number of predictions made by the classifier divided by the total number of all observations (see Table 3). For instance, if a model was exact 80 times from 100 cases, the accuracy could be viewed as 80 %. The misclassification rate is calculated using the number of incorrect classified observations or by one minus the accuracy.

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions} \tag{17}$$

Moreover, the accuracy of a classifier is defined as the fraction of correct predictions on test set. It gives assessment of correct predictions probability:

$$Accuracy = \frac{1}{n} \sum_{i=1}^{n} I(y_i = \hat{y}) \tag{18}$$

Where y_i denotes actual class, $\hat{y}$ indicates predicted class of y_i. I is an indicator function which has 1 for its true argument, and otherwise is zero.

F1-Score(F-beta) tries to balance and combine both recall and precision rather than using them individually. It is useful in some cases when the decision is required to choose the best performance of models. The maximum value of the F1-Score is 1 for a perfect classifier. The general definition of f-beta is:

$$F1\ Score = (1 + \beta^2)\ \frac{2 * Precision * Recall}{(\beta^2 * Precision) + Recall} \quad (19)$$

β is the weight of precision in harmonic mean. Various values of β give different value of weight to precision and recall. The greater β values are required. The most often used performance measurements [99] for binary and multi-class classification are provided in Table 4.

Table 4: Measures for binary and multi-class classification [103]

Binary Classification Measures		Multi-class classification	
Performance Measure	Formula	Performance Measure	Formula
Accuracy	$\frac{TP + TN}{TP + FP + TN + FN}$	$Accuracy_M$	$\frac{\sum_i^l \frac{TP_i + TN_i}{TP_i + TN_i + FP_i + FN_i}}{l}$
Error rate	$\frac{FP + FN}{TP + FP + TN + FN}$	$Error_M$	$\frac{\sum_i^l \frac{FP_i + FN_i}{TP_i + TN_i + FP_i + FN_i}}{l}$
Precision	$\frac{TP}{TP + FP}$	$Precision_M$	$\frac{\sum_i^l \frac{TP_i}{TP_i + FP_i}}{l}$
Recall (Sensitivity)	$\frac{TP}{TP + FN}$	$Recall_M$	$\frac{\sum_i^l \frac{TP_i}{TP_i + FN_i}}{l}$
F-Score	$\frac{2 * Recall * Precision}{Recall + Precision}$	$Fscore_M$	$\frac{2 * Recall_M Precision_M}{Recall_M + Precision_M}$
AUC	$\frac{1}{2}(\frac{TP}{TP + FN} + \frac{TN}{TN + FP})$		
Specificity	$\frac{TN}{TP + FN}$		

2.3.4 Classifier evaluation

The evaluation procedure estimates how well a model can generalise to out-of-sample data in practice [55]. We describe two different techniques used to split up the training dataset to create useful estimates of performance for classification algorithms:

a) Train and test datasets

The standard method of evaluating a classifier is to use different datasets of training and testing [28]. Typically, a given data set is randomly divided into a disjoint train dataset and test dataset.

Train dataset is utilised for training the classifier on the first part, and on the second part, the test dataset is used to assess the performance of the classifier against the expected results [19]. The typical size of the split to use 70% of the data for training and the remaining 30% for testing. When the used algorithm is slow to train, the algorithm evaluation process can be fast utilising this approach. However, using this technique can generate a high variance. Thus, meaningful differences in the estimate of accuracy between the training and test dataset can result. For calculating the performance of the learned model, the hold-out strategy can be utilised to split the dataset into two sections, i.e. train dataset and test dataset. The performance assessment achieved by held-out approach relies on the division of training and testing data. The evaluation can be carried out many times, and the average is calculated [18].

b) K-fold cross-validation

The performance of the learned classifier gained via the method of training and testing is evaluated in the classification process. One commonly used method for training and testing the learned model, i.e. cross-validation [28][102]. Cross-validation is a common approach to compute the expected value of the performance of classifiers. Cross-validation splits a given dataset into k equal size folds (k parts) randomly. Each fold is treated as a testing dataset ($fold_i$). After training the classifier on the remaining folds ($folds \backslash fold_i$) represents the training dataset. We evaluate its performance θ_i on the test dataset ($fold_i$) and report the mean and the variance θ for the error rate as follows:

$$\theta = \frac{1}{k} \sum_{i=1}^{k} \theta_i \tag{20}$$

And its variance as:

$$\sigma^2 = \frac{1}{k} \sum_{i=1}^{k} (\theta_i - \theta)^2 \tag{21}$$

We can repeat the whole cross-validation approach multiple times. Next, the average of the mean of error rate can be computed.

c) Comparing classifiers by Paired t-test

This method can be applied to report any significant differences between individual classifiers for comparing classifiers to check the difference in the results of two classifiers [28][103]. This method is used to estimate which of the classifiers on a given dataset has a superior classification performance. Consider a given dataset described in Table 5.

Table 5: Datasets summary

Dataset	Description
D1	Pima Indians Diabetes
D2	Breast-cancer dataset
D3	Indian Liver Patient data
D4	SPECTF Heart dataset
D5	Thyroid dataset

For the datasets described in Chapter 3, we built classifiers on identical datasets. The classifiers are trained and tested on the same data. Let θ_i^{RF} and θ^{NB}_i indicate the values of the error rate measure for random forest (RF) and naive Bayes (NB) classifiers, respectively. We want to assess the difference in the classifier's performance on the same dataset. This method is described in [28], which is for comparing classifiers by using Paired t-test to assess the difference in the classification performance of two classifiers. We perform the hypothesis test to investigate this problem. The null hypothesis $H0$ is that the classifiers are not different, whereas the alternative hypothesis Ha is that they are different. To determine if the two classifiers are different or not different based on the difference between their performance.

$$H_0 : \mu_{dif} = 0 \qquad H_a : \mu_{dif} \neq 0$$

Where, H_0 and H_a are mathematical opposites.

The difference in the classifier's performance on the same dataset

$$dif_i = \theta^{NB}_i - \theta^{RF}_i$$

The mean of the difference can be calculated as:

$$\hat{\mu}_{dif} = \frac{1}{K}\sum_{i=1}^{K} dif_i$$

The variance is computed as:

$$\hat{\sigma}^2_{dif} = \frac{1}{K}\sum_{i=1}^{K}(dif_i - \mu_{dif})^2$$

$$\hat{\sigma}_{dif} = \sqrt{\hat{\sigma}^2_{dif}}$$

$$t_{dif} = \frac{\hat{\mu}_{dif} - \mu_{dif}}{\frac{\hat{\sigma}_{dif}}{\sqrt{K}}}$$

we obtained the following results:

Table 6: The error rates and the difference over each of the k=10 folds for Pima dataset (D1)

	1	2	3	4	5	6	7	8	9	10
θ^{RF}	0.208	0.208	0.276	0.169	0.273	0.221	0.169	0.247	0.197	0.286
θ^{NB}	0.403	0.197	0.211	0.195	0.221	0.234	0.221	0.299	0.182	0.221
dif_i	-0.195	0.011	0.065	0.026	0.052	-0.013	-0.052	-0.052	0.015	0.065

$$\hat{\mu}_{dif} = \frac{-0.078}{10} = -0.0078$$

$$\hat{\sigma}^2_{dif} = 0.006$$

$$\hat{\sigma}_{dif} = \sqrt{0.006} = 0.077$$

$$t_{dif} = \frac{-0.0078}{\frac{0.077}{\sqrt{10}}} = -0.32$$

For the level of confidence, C $= 0.95$ or also called the level of significance ($\alpha = 1 - C$) and K − 1 = 9 is the degrees of freedom (df), we have $t_{K-1} = 1.833$ which is computed by using t - table [28]. The test statistic is used to conclude this problem. We can estimate where this value lies in the curve $t_{dif} = -0.32 \in (-1.833,\ 1.833) = (-t_{K-1}\ ,\ t_{K-1}\ )$. If this t_{δ}value falls in the rejection region, that means we can reject the null hypothesis $H0$. According to the result, we can reject the performances of the classifiers are the same and accept the alternative hypothesis Ha.

The result is there is no significant difference between the naive Bayes (NB) with the RF classifiers for this dataset. However, the results for the five datasets are shown in Table 7.

Table 7: The results on datasets D1 to D5

The parameter	D1	D2	D3	D4	D5
$\hat{\mu}_{dif}$	-0.0078	-0.0102	-0.0811	-0.0857	0.0176
$\hat{\sigma}^2_{dif}$	0.006	0.001	0.007	0.008	0.007
$\hat{\sigma}_{dif}$	0.077	0.032	0.084	0.089	0.084
t_{dif}	-0.32	-1.008	-3.053	-3.045	0.663

The result is that there is no significant difference between the naive Bayes (NB) and random forest classifiers for D1, D2, and D5 dataset. The value of t_{dif} does not fall in the rejection region (fail to reject null hypothesis H_0) as shown in Figure 6. However, t_{dif} for D3 and D4 do fall in the rejection region (reject null hypothesis H_a). These results mean there is a difference between the classifier's performance on these datasets. Using the error rate as the performance measure, we perform the values for the error rates and their difference over each of the ten folds on the five datasets (see Table 8).

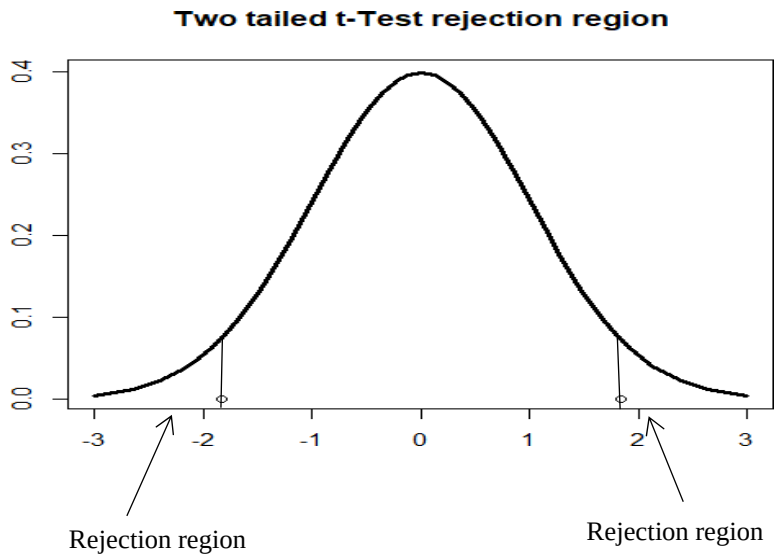


Figure 8: Two tailed t-Test rejection region

Table 8: The error rates and the difference over each of the k=10 folds for all datasets

		1	2	3	4	5	6	7	8	9	10
D1	θ^A	0.208	0.208	0.276	0.169	0.273	0.221	0.169	0.247	0.197	0.286
	θ^B	0.403	0.197	0.211	0.195	0.221	0.234	0.221	0.299	0.182	0.221
	dif_i	-0.195	0.011	0.065	0.026	0.052	-0.013	-0.052	-0.052	0.015	0.065
D2	θ^A	0.058	0.029	0.015	0.043	0.015	0.015	0.030	0.014	0.029	0.015
	θ^B	0.029	0.015	0.044	0.029	0.059	0.000	0.044	0.072	0.029	0.044
	dif_i	0.029	0.014	-0.029	0.014	-0.044	0.015	-0.014	-0.058	0.000	-0.29
D3	θ^A	0.228	0.339	0.281	0.316	0.241	0.254	0.293	0.310	0.271	0.316
	θ^B	0.397	0.254	0.397	0.397	0.333	0.373	0.351	0.316	0.492	0.368
	dif_i	-0.169	0.085	-0.116	-0.063	-0.092	-0.119	-0.058	-0.006	-0.221	-0.052
D4	θ^A	0.111	0.143	0.192	0.231	0.214	0.192	0.185	0.192	0.115	0.148
	θ^B	0.296	0.296	0.115	0.231	0.385	0.154	0.250	0.370	0.269	0.214
	dif_i	-0.185	-0.153	0.077	0.000	-0.171	0.038	-0.065	-0.178	-0.154	-0.066
D5	θ^A	0.067	0.276	0.200	0.167	0.172	0.241	0.200	0.133	0.233	0.133
	θ^B	0.233	0.167	0.200	0.233	0.103	0.138	0.138	0.167	0.167	0.100
	dif_i	-0.166	0.109	0.000	-0.066	0.069	0.103	0.062	-0.034	0.066	0.033

Figure 8 illustrates that the null hypothesis in the tails is rejected and do not rejected in the middle.

2.3.5 The classifiers quality metrics

In this section, the quality metrics of classification models are discussed about several criteria. The meaning of the evaluation metrics should be clear for experts or users for selecting an optimal classifier [103]. Moreover, the selection of the parameters is chosen based on the objective of them. The metrics may be conflicted or fuzzy, which need a way of treatment in the evaluation process. For example, the accuracy and the comprehensibility are distinguished as two objectives to be minimised in [103][101], and only the complexity in [104].

Table 9: Quality meta-metrics of a classification model

N	Quality Metrics	Evaluation metrics	Description
1	Correctness	Accuracy	percent correct, precision, recall, F measure
		error metrics	percent incorrect, FPR, FP, TN
2	Complexity	Computational	Elapsed Time training, User CPU Time training
		Memory/Space	NumRules, Tree Size, NumLeaves
3	Responsiveness	Responsiveness	Elapsed time testing, UsrCPUtime testing
4	Consistency	Consistency	Standard deviation
5	Reliability	Information-Theoretic	Entropy, entropy gain
		Distance or Error Measure	MAR, RMSE
6	Comprehensibility	Comprehensibility	Measures Interestingness and Interpretability, e.g., Num. Rules, Tree Size etc
7	Robustness	Robustness	Measure sensitivity in terms of True positive rate
8	separability	separability or coherency	AUR, ROC

Several commonly used evaluation criteria can be used to evaluate classifiers, such as the accuracy, ROC curves and RMSE [105]. Ali et al. [103] defined that quality meta-metrics (QMM) for the classifier's evaluation metrics. A list of available parameters is given in table 8 below, which consider families of classifiers such as decision tree. The performance of an optimisation approach can be measured on some criteria[106], which are given in Table 9.

We consider in this thesis number of criteria, namely robustness, reliability and correctness evaluation metrics of classifiers. Evaluating the correctness of classification can be performed by calculating the amount of correctly recognised class examples (true positives), (true negatives), (false positives) or (false negatives). These four counts are a confusion matrix for the binary classification situation shown in Table 3.

This section presented a background on classification as supervised learning. Section 3.2 provides classification algorithms, ensemble learning for classifiers, the popular classification algorithms and the evaluation procedure. The list of the evaluation metrics of classification models is provided, which can be useful for experts or users for selecting an optimal classifier. Related work is presented with consideration of the robustness, reliability and correctness evaluation metrics of classifiers. We provided an example of using the Paired t-test for finding the difference between classifiers performances.

2.3.6 Studies of robustness in machine learning

In the machine learning literature, robustness is an essential property to deal with massive amounts of data that are not subject to any quality control. In classification and regression problems, efficient learning algorithms have been proposed to obtain a ''good'' outcome. Table 10 shows some research work concerning the robustness of the classification model.

The purpose of a trained classification model is to classify new instances from the given domain. Classifiers evaluation refers to assess the quality of the outcomes represented by the model. Previously, many studies made to evaluate the performance of the classification models in many domains. Machine learning is increasingly used in various domains, such as healthcare informatics. Recent example a study presented by Oude et al. [101] for exploring the possibilities of using supervised machine learning in the design of a clinical decision support system (CDSS) to support patients with low back pain (LBP) in their self-referral process to primary care [101]. LBP can cause human physical disability, which prevented many people at an early age from engaging in daily work and activities. A comparison of the three classification models, namely decision tree, random forest, and boosted tree was performed to assess the performance of the classifiers and then decide about

the best classifier to use in real practice. Conclusions of this study showed promising outcomes on the use of machine learning in CDSS design. The boosted tree model provided the best performance to classify low back pain instances. However, it still needs to be enhanced. Particularly cases that are categorised as self-care instances.

Another study in [77] provided a general comparison with state-of-art machine learning algorithms. This work addressed the effectiveness of supervised machine learning algorithms regarding the accuracy, speed of learning, complexity and risk of overfitting measures. Bittencourt et al. [107] presented an approach to identify the changed areas caused by fire. The research introduces some appropriate models of classifiers, including random forest and an ensemble model, resulting in productive outcomes. The developed approach is validated throughout the region of Brazil's Woody Savannah against reference data obtained from expert manual classifications. More information from distinct areas will eventually be used later, depending on the results of the techniques used.

Specialists had been building computer programs for Amazon in 2014 [108]. This artificial intelligence tool of recruiting prefers men for technical jobs. Computer program had been established since 2014 for Amazon to check resumes (CV) of candidates and specify scores (1-5) for job candidates. However, in 2015, the ML specialists in the company recognised that the ranking generated by the system of technical jobs was not neutral in term of gender. There is no diversity and equality of the outcomes. The reason for this big issue is due to building the model based on data that collected over ten years period of resumes submitted to Amazon. In that period, most were male candidates. Thus, the system prefers men because most of the candidates were men. The model relied on this imbalanced data to generate the ranking of the candidates [108]. It seems the model could not consider all present data.

Pelletier et al. [109] attempted to assess the robustness of random forests to map land cover with a satellite image. Data of satellite image given by High spectral, spatial and temporal Resolution Satellite Image Time Series. However, there are some challenges of adapting traditional classification schemes to data complexity. These challenges include:

- Determining which classifier can handle the variability of data.
- Dealing with a significant amount of data.
- Choosing the best feature set used as input data.
- Finding the trade-off between classification accuracy.

The RF classifier has produced equivalent results to the SVM method with a better trade-off between the classification performances and the computing times. Moreover, when input features changed,

they showed fewer distinctions in terms of accuracy. Therefore, RF is an appropriate tool for managing the quantity of data supplied by HR-SITS [109]. SVM and RF demonstrate some complementarity primarily for low accuracy classifications. The combination of both classifiers might result in more accuracy outcomes than a single classifier for these classifications.

Table 10: Some research work concerning the robustness of the classification model.

Reference	ML Methods	domain	Data set	results
Shami and Verhelst [33]	K-nearest neighbors (KNN) Support vector machines (SVM) Ada-boosted decision trees	emotional speech databases	Kismet, BabyEars, Danish, and Berlin databases.	robust classification outcomes on the integrated databases
Pelletier et al. [112]	Random Forests (RF) Support Vector Machines (SVM)	remote sensing sensors	High spectral, spatial and temporal Resolution Satellite Image Time Series (HR-SITS)	Accuracy 83.3 % for RF, and 77.1 % for SVM.
Kanamori et a. [45]	Support Vector Machines (SVM)	non-convex optimization problem	Synthetic data	The optimal local solution of ML algorithms has the robustness property
Liu et al. [113]	feature extraction and selection methods	mobile app traffic	deploying the mgtClient on 10 volunteers' smartphones	selecting feature subset has improved the robustness of mobile app traffic classification

Another approach to assess the robustness of learning algorithms is introduced by Kanamori et al. [40]. This approach depends on using hinge loss with outlier indicators. Outliers could increase bias.

They proposed a merged formulation of assessing robustness property of classification and regression learning methods based on evaluating the breakdown point. Some statistical characteristics from the standpoint of robustness measured by influence function, gross error sensitivity, or breakdown point. Wang et al. have proposed an approach for analysing the robustness property for classification and regression problems by studying the robustness property of the optimal solution of used learning algorithms [110]. They demonstrated an integrated approach between obtain optimal feature sets and feature extraction to improving classification robustness of mobile. Feature extraction is a method of dimension reduction that decreases the number of features required for processing without losing valuable data. Feature extraction method can improve learning pace and generalisation steps in the process of machine learning [111]. Another study is an evaluation of the robustness of the existing supervised machine learning approaches to the classification of emotions in speech [31]. KNN, SVM and Ada-boosted decision trees are considered. Moreover, four emotional speech databases are used, Kismet, BabyEars, Danish, and Berlin databases. They constructed ML classifiers on the integrated databases, and this provides promisingly robust classification outcomes, indicating that emotional corpora with emotion classes recorded under distinct circumstances can be used to build a single classifier capable of distinguishing feelings in the merged corpora. Robustness and prediction accuracy of machine learning for objective visual quality assessment can be found in [112].

2.4 Applicabilitydomain

This section presents the applicability domain concept under three heading: introduction, methods of estimation applicability domain, and applicability domain and machine learning. Defining the applicability domain of the machine learning model is an important task and sometimes result in inefficient performance [113]. The reason may include lack of knowledge about the capability of the model. The applicability domain is defined as the ability of the model to determine whether new data satisfies the assumptions of the model [114]. The level of generalisation of a given predictive model can be determined by defining its applicability domain AD. In this way, if the AD is too restricted, it means the model expectations can be very limit. According to [115], the Quantitative Structure-Activity Relationship (QSAR) model should have a definition of applicability domain (AD) and appropriate measures for goodness-of-fit. Even though some models have high accuracy as carried out in many studies [115], it is useful to determine where the model can provide reliable results [4].

Pharmaceuticals is one of industry-specific data mining. One of the challenges of data mining of pharmaceutical information is related to predicting safety issues and the whole procedure of drug discovery. Predicting safety issues is the entire process of drug discovery, extensive collections of data made concerning both the desirable and undesirable properties of drugs or drug candidates [88][116][117][118]. The process of drug discovery is accelerated by using predictive models to complement or as an alternative to physical safety testing. These models are used to prioritise research directions and avoid taking drug candidates with potential problems further [117]. Collecting and normalising the data is challenging since the chemicals may have been tested using different types of experiments or experimental protocols. The results are often obtained from controlled trials, generated for a specific variety of chemicals. To use this data to make predictions concerning the general population of possible chemicals requires care in putting the training sets together [114]. This training set should ideally now represent a diverse set of chemicals to increase the applicability of any predictive model generated. The types of chemicals in the training sets limit what kinds of chemicals can be provided as input to models.

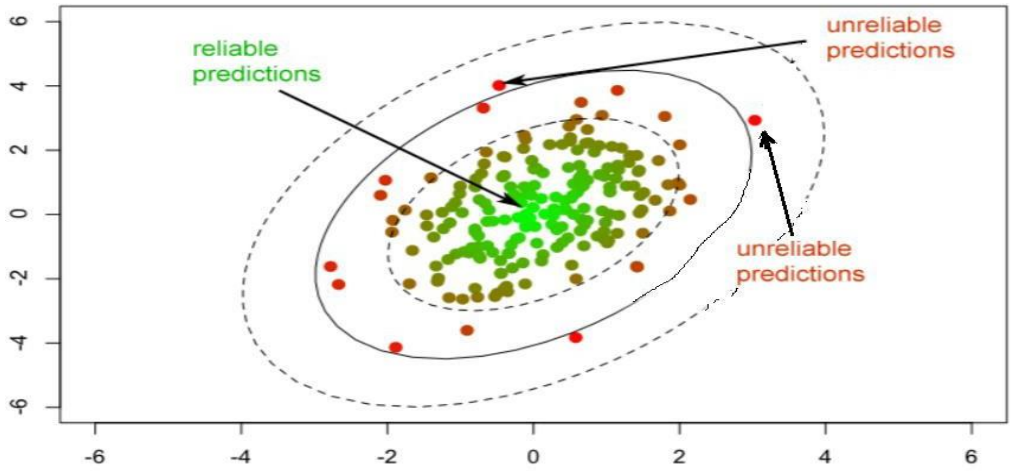


Figure 9: Dissimilarity to the training set [109]

It is usual to assess whether a specific compound can be used with a model by comparing the chemical to be tested against the training set of the model, as shown in Figure 9. When the compound to be predicted is outside this applicability domain, a prediction would not be reliable [119]. Figure 9 illustrates dissimilarity to the training set [120]. The domain of applicability can be recognised as all cases with AD below a specific [120].

It is necessary to assess the predictive performance of the classifier during model development. This can be done by testing the classifier on new data set to estimate the prediction error. But what if the original dataset is not like the training dataset. Therefore, the prediction could be with significant error because the classifier is applied in the uncovered domain by the training set. In this case, some data

are required with specific characteristics of a given dataset to test AD of the model [121]. In the following section, an explanation of how to generate some synthetic data that can help to perform AD assessment. The methods for determining AD are reviewed in fig. Among these existing ways, no technique can be considered as the best method. Each approach can have advantages and disadvantages [122].

2.4.1 Methods to estimate the applicability domain

Theoretical models of quantitative structure-activity relationships (QSARs) relate a quantitative measure of chemical structure to a physical property or a biological impact. QSAR predictions can be utilised for chemical risk assessment for the protection of human and environmental health, which makes them attractive to regulators, especially in the absence of experimental data [123][124]. There are many approaches used to assess the applicability domain of the QSAR models in multivariate space [124]. The existing strategies for defining AD of QSAR models are for regression and classification models, as illustrated in Figure 11. The most used approaches for estimating AD include the followings.

2.4.1.1 Range-based methods (or geometric methods)

We can identify the applicability domain of models by determining the region where the data are in the space. This process includes defining borders of the area, which tacked into account the description of each attribute of the dataset. Hyper rectangle is identified from the minimum and maximum values of each feature used for building the model. However, this approach can be insufficient for a large data set. Therefore, the technique of dimension reduction can be used to select appropriate features, such as principal component analysis (PCA). The PCA transforms the original data by axis rotation into a new orthogonal coordinate system. Newly formed axes are defined as PCs showing the maximum variance of the total dataset. Bounding Box considers the variety of descriptors used for model construction. The applicability domain can be described of a feature space distribution as a bounding box [125], which is an n-dimensional hyper-rectangle identified by the maximum and minimum values of each feature used to build the model (see Figure 10). The AD in the bounding box space, where the training set is the green circle (shown in Figure 10). The predictions within the sphere of the test set are regarded as reliable. When the test set is outside the model space will be less reliably predicted. Another method is called the convex hull [126][127]. The convex hull approach is defined as the smallest convex area, which contains the whole training dataset. The convex hull method works based on obtaining the domain of applicability of the smallest convex region of descriptor ranges that includes the training set.

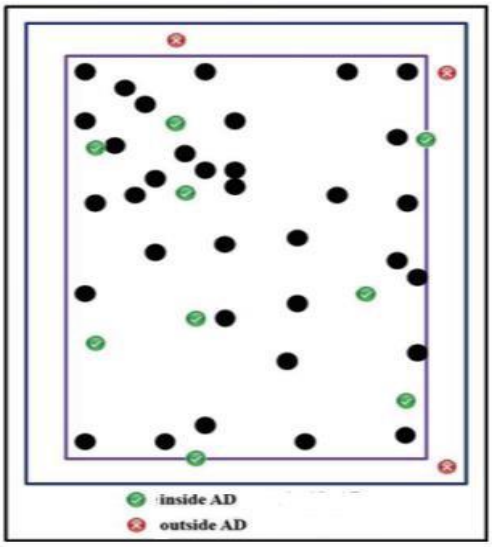


Figure 10: Descriptors of boundingbox [128]

2.4.1.2 Probability density distribution-based methods

They are an appealing way of estimating the reliability of model predictions [121]. Probability density distribution is another approach used for determining AD. This approach is divided into parametric and non-parametric approaches [128]. Parametric methods use the probability density function p(x) of standard distributions such as Gaussian and Poisson distributions. On the other hand, non-parametric techniques allow estimating the probability density from the data distribution. It is called a distribution-free method. Therefore, it has the capability of identifying internal empty regions inside the convex hull.

Further, the empty regions that are close to the convex hull border, this approach generates concave areas for reflecting the actual distribution of the data [124][128]. Among the existing methods, there is no best universally way [124]. Thus, the chance of uncertainty still related to the assessment of AD. If the built model is not reliable, one cannot get confidence in the AD assessment. However, estimating AD can be affected by some issues such as the dimensionality, data descriptors, response value as endpoints, data distribution, and the used algorithm of the AD determination process [122].

In addition to the AD methods mentioned above, several other approaches to defining the AD of QSAR models have been published in the literature. Some of which are mentioned in this part. These

methods are decision trees and decision Forests Approach [114], and Stepwise Approach to Determine Model's AD [129].

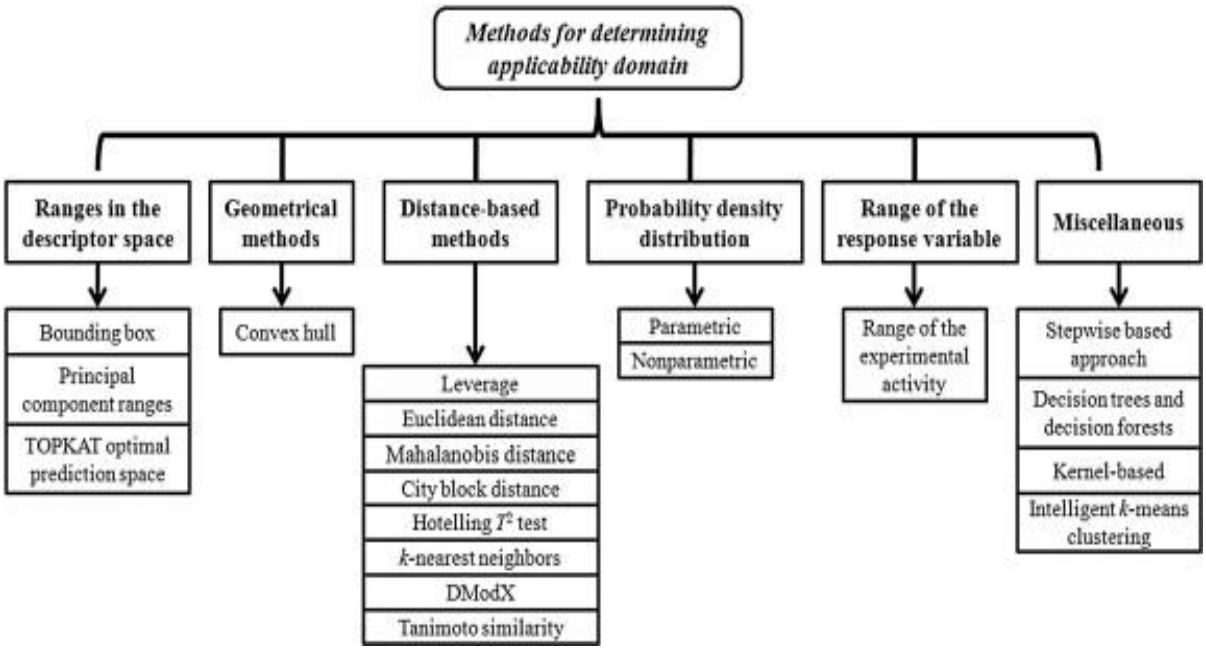
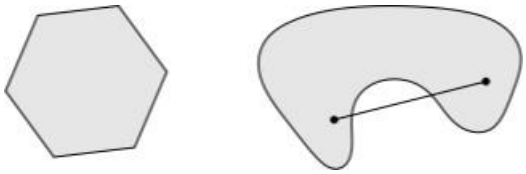


Figure 11: Classification of the AD approaches under different hypothesis [127]

2.4.2 Convex Hull

This section describes the convex hull as one of the methods of identifying AD. In computational geometry topic, the issue of reconstructing a set from a finite set of points has been treated of different fields of research. For example, computing convex hulls for finite dataset has essential applications in some domain such as pattern recognition, cluster analysis, computer graphics, robotics and image processing. The convex hull of the dataset S in the space is defined as the smallest convex polygon, which encloses all the points within S. It is a shape of bounding the points S. There is a way to define whether a polygon is convex or not. The mathematical definition is to join any two points $p, q \in S$ lie within the polygon. This line $\overline{pq}$should completely place in the polygon as well. Thus, $conv(S)$ is the straight-line segment (p, q). A considerable amount of literature has been done about the convex hull. There are some algorithms used forcomputing the convex hull problem are developed such as Graham scan- $O(n\ log\ n)$ [130] and Gift wrapping, Chan's algorithm and Jarvis's march - $O(h.\ n)$,which h is the complexity of the convex hull [131][132][133]. Moreover, the convex hull can be computed using dived-and-conquer approach developed by Preparata and Hong [134]. The algorithm generates segments of the convex hull by steps. First, break the points up into two sets

right and lift. Next divide these two subsets. Further, in the same way. Then upper tangent and lower tangent will be obtained. Discard all the points between two tangents to make the convex hull [134].



(a) Convex hull (b) Not convex hull

Figure 12: Examples of simple convex and nonconvex sets [129]

The output of the convex hull is a sequence of points in clockwise order. It is important to distinguish between convex shape and concave shape [135]. Concave includes angle is greater than 180 degrees.

Practically, a set is convex if every point in the set can be seen by every other position, along a clear straight line between them. The set is considered as convex, as it includes the entire line between any two separate points, and hence the line segment between the points. Figure 12 shows two simple sets (a) and (b) of convex and nonconvex in R^2, respectively. The hexagon in (a) includes its boundary is convex. The kidney-shaped set in (b) is not convex since the line segment between the two points in the set (shown as dots) is not contained in the set [126].

Figure 13 illustrates the definition of the convex hull for data points in 2D space. However, the convex hull can bd defined in any dimension [134]. The convex hull in multiple dimensions can be computed in $O(n \log n)$ time. The example of convex hull in Figure 13 is achieved in the R language. There is a straight line between any two points within the polygon.

Definition1: Convexity a set $S \subseteq R^n$ is convex for any $p \in S$ and $q \in S$ implies that the segment $\bar{pq} \subseteq S$.

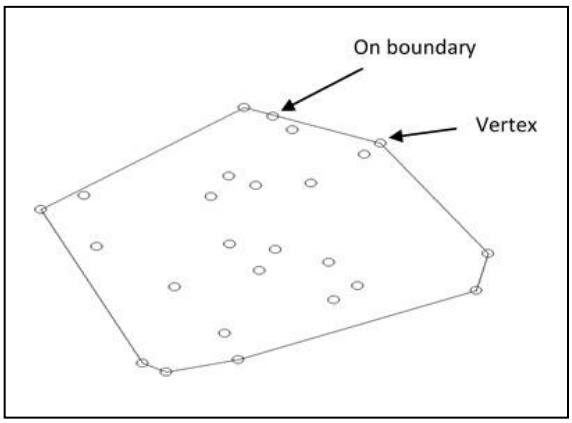


Figure 13: Example of Convex Hull of a set of points in 2D space

Definition 2: The convex hull $conv(S)$ of a set $S \subseteq R^n$ is the intersection of all convex supersets of S.

Given a set of points S in plane. $S = \{S_i = (x_i, y_i), i = 1,2, \dots, n\}$. S is a subset of R^n, For any $S \subseteq R^n$, we have $Conv(S) = \{\sum^n_{i=1} \lambda_i S_i \mid n \in N \wedge \sum^n_{i=1} \lambda_i = 1 \quad \wedge \forall i \in \{1, \dots, n\}: \lambda_i \geq 0 \wedge S_i \in S\}$.

The convex combinations of S are represented by the elements of the set on the right-hand side.

Convex hull of S is the smallest convex polygon which cover all the points, and all the points are bounded with this polygon. It is intersection of all convex sets which contain S. $Conv(S)$ denotes the convex hull of a set S. The polygon is not convex hull if the connecting line segment of some pairs of points is not entirely contained within the polygon. Note that for any point $(S_i)_{i \in I}$ of convex sets, the intersection $\bigcap_{i \in I} P$ is convex [136]. This section presents the computation of the convex hull of a set of points in the plane. There are some the fundamental theorems about convex sets in R^2:

Theorem1 ([137]): take a collection $Conv = \{Conv_1, \dots C_{onvn}\}$ of $n \geq n + 1$ convex subsets of R^2, such that any $d + 1$ pairwise distinct sets from $Conv$ have non-empty intersection. Thus, the intersection $\bigcap^n_{i=1} Conv_i$ of all sets from $Conv$ is non-empty.

Theorem 2 ([132]): any set $p, q \subset R^d$ of $d + 2$ points can be partitioned into two disjoint subsets p and q such that $Conv(p) \cap Conv(q) \neq \emptyset$.

Theorem 3 (Separating Hyperplane Theorem) //: any two compact convex sets $Conv_1, Conv_2 \subset R^d$ with $Conv_1 \cap Conv_2 = \emptyset$ can be separated by a hyperplane. There exists a hyperplane H. $Where, H \neq 0, H \in R^d$ such that $Conv_1$ and $Conv_2$ lie in the opposite open half spaces bounded by H (See Figure 12).

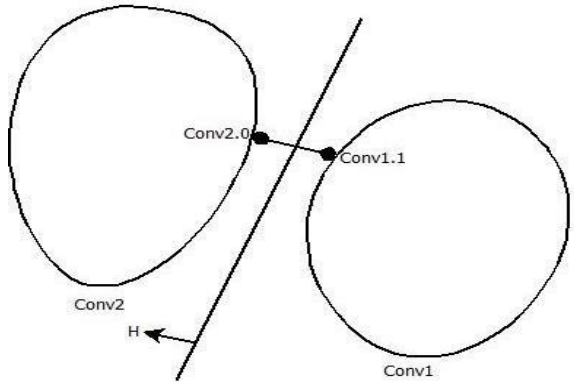


Figure 14: Separating Hyperplane Theorem [140]

Taking int consideration the distance function $\delta: Conv_1 \times Conv_2 \rightarrow R \; with \; (Conv_{1.0}, Conv_{2.0}) \rightarrow ||Conv_{1.0} - Conv_2||$. At some point the distance function δ reaches its minimum $(Conv_{1.0}, Conv_{2.0}) \in Conv_1 \times Conv_2 \; with \; \delta(Conv_{1.0} - Conv_2) > 0$. Suppose H denotes the

hyperplane to the line segment $Conv_{1.0}Conv_2$ and crossing the midpoint of $Conv_{1.0}$ and $Conv_2$. Consider $Conv_1$ has points on both sides of H, then by considering the convexity of $Conv_1$, it has also a point on H, but we just saw that there is no such point. Therefore, $Conv_1$ and $Conv_2$ must lie in different open half spaces bounded by H [126].

The algorithm of constructing the convex hull described in this section is Graham Scan [138]. We focus on the problem in R^2 where the convex hull of a finite point set forms a convex polygon (see Figure 13 [134]). This algorithm is called Successive Local Repair due to its way to work. It begins with some polygon that encloses all points and then step-by-step repairs this polygon by removing non-convex vertices. It works in two phases:

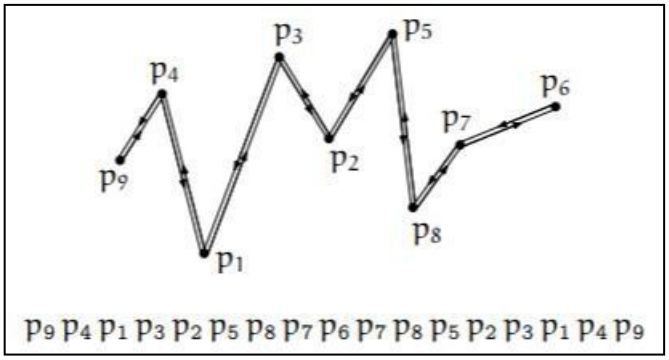


Figure 15: Sorting points in Graham Scan [137]

Phase1: sort the points lexicographically and excluding duplicates $(p1, \ldots, pn)$ as shown in Figure15. Phase1: keep track of three points and find the angle formed by them. Then reject points from the sequence if there is a (sequent) triple (p, q, r) such that r is to the right of the line.

Theorem 4 The convex hull of a set $P \subset R^2$ of n points can be computed using $O(n\ log\ n)$ geometric operations. Graham Scan algorithm uses $O(n\ log\ n)$ geometric operations.

2.4.3 Applicability domain and machine learning

There are attempts to estimate the applicability domain of kernel-based machine learning models in different fields [5]. In this section, the details of the studies providing applicability domain with machine learning algorithms are explained.

Machine learning techniques such as kernel-based machine learning methods have become a popular technique for learning QSAR models. However, other methods like neural networks or decision trees, estimate AD based on the structure of the training set in the feature space [139]. Applicability domain description is shown in Figure 14 [139], orange data points represent a training set used for building a model. For new compounds (blue points) which place into the inner, the darker area is close enough to the training set; thus, the model can be applied with a level of confidence. However, new compounds which are not in the dark area are different from the training set. Therefore, the model

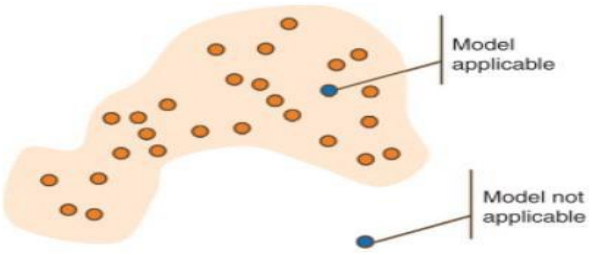


Figure 16: Applicability domain description [141]

should no longer be applied. When the new compounds are dissimilar to the training set compounds, it is not sure that the outcomes from the model are correct.

The AD is defined as the similarity of the structure of the instances. Moreover, AD is the similarity measurement depends on descriptors. The similarity is between the new compounds and the test set compounds. Tetko et al. has stated that both of these approaches using the analysis process for logP [140]. Study by Schroeter et al. outlined the diversity of methods for assessing the model applicability for new compounds [121]. The study is included the following techniques: range-based, distance-based methods, probability density distribution-based methods, ensemble methods and Bayesian methods. Another study by A.Palczewska using Pareto points for model identification in predictive toxicology [1].

Some studies investigating AD have been carried out on the chemical and biological sciences. Weaver in [141] defined AD as an essential task in quantitative structure-activity relationships (QSAR) for estimating the uncertainty in the prediction based on the similarity of the compounds used for building the model. Applicability domain of a (Q)SAR models is knowledge or information on which the training set of the model has been developed and is applicable to make predictions for a new dataset. Broadly, QSAR modelling is practised in various disciplines including industry, academy and governments in the whole world. So far, AD has only been applied to QSARs models to identify the region in chemical space where the model provides reliable predictions [124].

Study by Roy et al. presented the applicability domain using the standardisation approach presented the approach of applicability domain using the standardization approach [124]. Further, it is an attempt to define the X-outliers of the training set for identifying the compounds that are outside the AD from the test set. This approach depends on the standardisation approach.

In recent research by Klingspohn et al. [88] define the applicability domain of classification and prediction models as the region in the input data points space where the model produces proper and reliable predictions. The estimation of an applicability domain requires knowledge of the training set, but in some cases, the AD cannot be provided before model training. According to the authors in [141] describe AD as a concept to assess uncertainty in the prediction based on the similarity between new predictions and building data. Model validation of the models requires defining the Applicability Domain (AD) of these models by using different approaches and based on the problem that we address. It is vital since the model should be able to provide some reliable predictions, especially for health care data, where the decision is related to people live. Aniceto et al. demonstrate a study for determining the ability of the model for predictive in the regions of chemical space for ensuring the reliability of new predictions [115]. Some models have a high accuracy as carried out in some previous work [116], but ignore the possibility of including in the data points space where the model cannot give reliable outcomes.

A helpful AD should connect between the predictive reliability in the training set and external dataset equivalently. Although the growing use of QSAR predictions approaches for several purposes, validation of predictive models is required and essential part of defining the applicability domain (AD) of the model. A heuristic decision rule was extracted using the approach of integrating data distribution and exploits KNN principle [109]. Researches in [40] investigated the ability of RF and SVM to classify HR-SITS. The focus of the method was Overall Accuracy (OA) and training time of both classifiers. Fjodorova and Marjana evaluating the AD of the neural networks in the case of predictive classification models [142]. The metric for the propagation artificial neural network model's AD evaluation is the Euclidean distances between an object and the neural network's corresponding neuron. The investigation into the coverage of training and test sets in the descriptors space was conducted for false predicted examples.

The dk-NN AD proposed by Sahigara et al. [143] uses the k-NN principle associated with the concept of adaptive kernel techniques in KDE to detect local neighbourhoods within the data. This AD method works based on selecting the appropriate number of k-nearest neighbours. It allowed identifying a smooth region of k values were the results remained unchanged, ensuring high robustness in the AD definition. This strategy enables local reliability to be mapped in the training set space across different

locations and thus allows areas to where the model has low reliability to be identified [115]. Comparison of Different Approaches to Define the Applicability Domain of QSAR Models was made by Sahigara et al. in [114]. The comparison analysis was performed for each dataset and approach implemented in this study. It considered the statistics of the model and the relative position of the test set with the training space.

Finally, the space characterisation varied depending on the implementation of the applicability domain strategy, algorithm complexity, while others related to the parameters of the algorithm used. AD methods have their strengths and constraints, and therefore, it is up to the model builder to choose for his model the most suitable application domain strategy.

2.4.4 Advantages and disadvantages of applicability domain

In this section, the pros and cons of the AD approaches are discussed. Assessing AD methods may are powerful, but they cannot come without risks. While these methods have many advantages, the disadvantages should also be considered. We are considering the methodology of the test set structure to be within (or outside) the Applicability Domain. Range-based and geometric methods are the simplest methods of defining the AD of the model [129]. However, some drawbacks can be associated with this approach. First, this approach can be insufficient for a large data set because only descriptor ranges of molecules are considered. Second, empty regions cannot be defined in the space. Third, the correlation between descriptors cannot also be considered [114]. Moreover, data complexity from the increase in dimensions can affect convex hull method [144]. The primary characteristic of these methods is their capacity to recognise the empty areas. Besides, the actual distribution of data can be reflected by generating concave regions around the space boundaries [142].

Finally, this chapter presented the AD approach in three main headings, including a definition of AD, methods of estimation applicability domain, and machine learning algorithms for the AD. Moreover, it gives a background of the scientific aspects of the applicability domain and related work. It is essential to understand the significant elements of the applicability domain and its relationships with machine learning. Several methods of the evaluation AD have been discussed. Although the current arrangements which provide AD assessment are limited, they are useful in achieving good results with QSAR models.

This study is to investigate the connection between the applicability domain approach and ML classification model performance. The usefulness of assessing the AD for the classification model is

assessed through the reliability and the robustness of classifiers. The information that was arranged in this chapter led to obtaining a useful overview of all the aims and methods of this study. Mainly, AD has only been applied to QSARs models to identify the region in chemical space where the model provides reliable predictions.

There are approaches to assessing the classifiers based on AD methods such as the STD [114] method and the k-nearest neighbours' density (dk-NN) [115]. In this work, this technique maps the outcomes of new examples in term of distance to the model space while considering the reliability of nearby training instances. Accordingly, here, we used a reliability measure that results from two distinct effects, bias and precision as explained in [115] for the classification model. This section discusses assessing the AD of ML classification model, and feature selection impact of assessing the AD of the ML model.

There are many studies related to the evaluation and the robustness of the ML model. However, those studies do not consider the concept of the applicability domain (AD) yet. The issue is that the AD is not often well defined, or it is not defined at all in many fields. part of this work investigates the robustness of ML classification models from the applicability domain perspective. A standard definition of applicability domain regards the spaces in which the model provides results with specific reliability.

2.5 Summary

This chapter presents a detailed review of the literature related to machine learning, classification as well as the applicability domain.

Overall, the use of machine learning algorithms to solve real-world issues is increasing. Therefore, this section highlighted the need for these methods to be evaluated. The potential in using machine learning to address many real-world problems, but it is essential to emphasise that many difficulties need to be resolved. The most frequently used quality criterion for classification issues is the accuracy, which can be assessed using the region under the ROC curve, misclassification rate. However, the use of accuracy measure only as the sole criterion of the ML model quality does not fully capture the demands of many apps in the real world. Various (maybe conflicting) measures need to be considered. Further, the applicability domain is recognised by research on machine learning evaluation.

From the literature review presented in the above sections, it is observed that most of the real work performed in the literature considered classification models evaluation. Hence, further investigation

is needed for into the classification evaluation considered the applicability domain of the classifier. While performing this approach, one challenge is missing values treatment in the pre-processing step. The method of handling missing values can affect the data in various forms. Currently, different processes like removing records (or fields) with missing values from the dataset and use of imputation methods are used to deal with missing values problem.

Based on the gaps identified by reviewing the literature, the topic of the classification model has gained more interest throughout the last years. Many of these models are useful at the same time required time and storage; thus, they may be reused and recycled. Practically, big data is not only a challenge, but the models as well became another dimension in significant data challenges. Assessing the AD of the model may be performed for explaining its outcomes, modifying or combining with existing knowledge.

This section provides an informative discussion based on literature reviews covering some aspects of machine learning, types algorithms for ML models. It has been structured to provide a thorough scientific understanding of the topic of this study, referencing the primary sources of information discovered during literature searches. We considered the three main types of machine learning, namely, supervised, semi-supervised, and unsupervised learners. Furthermore, preparing data steps are described in this section. Before beginning the work on a machine learning project, we discovered the significance of defining the data set. We have described various methods to prepare a dataset. Feature selection methods are used in the data preparing stage — finally, the use of machine learning, especially in the healthcare domain. The various techniques frequently used for pre-processing data stage of building the ML model were discussed in this chapter.

It should be noted that these methods represent the most techniques used in the pre-processing of the data, and it is challenging to tackle all the ways in a single section. The most common techniques used in the preparation of the data are cleaning the data, transformation, feature selection, and dealing with missing data [18]. The FS plays a part in data processing, removing redundant and meaningless features. In this section, each of these techniques will be briefly discussed.

3 Methodology

3.1 Introduction

Generally, machine learning algorithms rely on datasets fed into algorithms to execute the learning task. Datasets and research methodology play an essential role in the conduct of a research study. In this section, we describe the dataset, the techniques and methods used, and the different methods used to implement the proposed approach. Section 3.2 gives the data set used to train, test and validate the algorithm for classification. This chapter also explains the issues with the dataset. Data preparation is presented in section 3.3. Section 3.4 describes the research methodology.

3.2 Datasets

Several databases could be used to test the performance of the method. This study experiments were performed to seven datasets. This work makes use of only publicly available datasets obtained from the UCI Machine Learning Repository [145] and Kaggle [146]. The description of the datasets used in evaluation the performance of the classification techniques is given in Table 9.

Table 11: Summary of datasets

Dataset	Description	Features	Instances	Class
1	Pima Indians diabetes	9	768	2
2	Breast-cancer dataset	11	699	2
3	Indian liver patient data	11	583	2
4	Heart dataset	14	303	2
5	Thyroid dataset	21	7200	3
6	Cardiotocographic dataset	25	2130	3
7	Hepatitis	20	155	2

The data have been used as a test case for the proposed algorithms. The reasons for employing these datasets for our research work are (a) they are well-known datasets to practice machine learning and investigate the applicability of proposed techniques, (b) they are also real-world datasets, and (c) well-studied for comparing the obtained results. Since all the data sets used have a reasonable number of

observations, they are divided into a training set (70%) and a test set (30%). The test set will stay unchanged throughout the analysing of the techniques.

Some issues related to the quality of the data that can complicate a learning task and reduce the accuracy performance of the trained classification model are missing or inadequate information [23]. The reader is referred to section 2.2.2. For the experiments, the programming language R is being used to perform different tasks. R is a free software environment for statistical computing [102]. It contains packages available in the CRAN package repository [147]. Some packages related to data mining are used in this study including `randomForest`, `party`, `mlbench`, `caret`, `fields`, `e1071`, `rpart`, `ggplot2 and tree` package. Related codes are shown in appendix B.

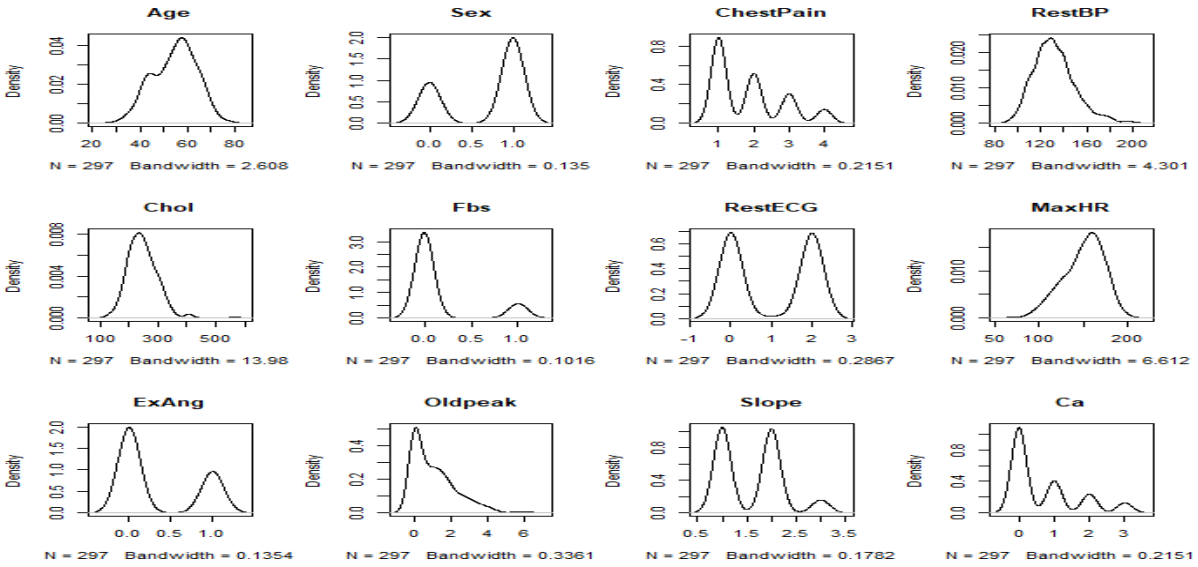


Figure 17: Density plots by attribute for Heart disease dataset

R is an open-source language for Machine Learning. However, the programming language one should choose for machine learning directly depends on the requirements of a given data problem, the preferences of the data scientist and the context of machine learning activities. R is a good choice to explore the data by using statistical methods and graphs. R has several machine learning packages for the many machine learning algorithms. UCI Machine Learning Repository website was used to download the data sets in Comma Separated Value (CSV) files. performing a general visualized overview of the data by using `summary ()` function to understand the data. some R machine learning classifiers require that the target feature is coded as a factor. Therefore, the target feature was transformed to factor by using `as. factor ()` function. However, converting input values to numeric can bd done by using `as. numeric ()` function. Overview of all the packages that are used in R can be found in [148]. `caret` is a package which includes a lot of algorithms. Some ML

algorithm can train a model with the `train ()` function. Visualization is a method to improve understanding of the dataset. It includes charts and plots from the data. Plots of the distribution of attributes for detecting outliers or invalid data. Plots of the relationships between attributes can help to reduce redundant attributes. In Figure 17and Figure 18, density plots by attribute for Heart disease dataset, and bar plot of each categorical attribute for Breast cancer dataset are visualized by using R packages. Density plot is useful for visualising an abstract of the distribution of each variable.

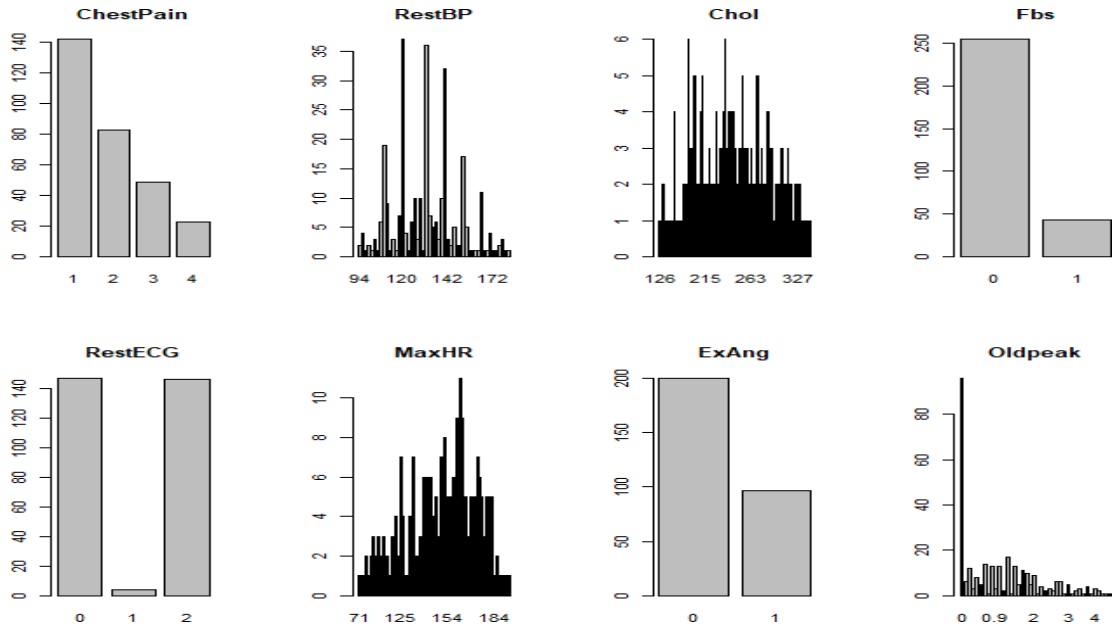


Figure 18: Bar plot of each categorical attribute for Breast cancer dataset

There are different univariate plots of individual attributes to learn about the distribution of each attribute such as histograms, and box plots

3.3 Data preparation

Preparing data is required to obtain the best results from machine learning algorithms (See section 2.2.2). In this study, preparation data is done to discover its structure to machine learning algorithms using in R packages such as caret package. Preparing data includes Data cleaning, transform data, and feature selection. Before fitting ML models, some preparation was needed to be done. Data pre-processing was achieved by:

- The optimisation was done with respect to Recursive feature elimination (RFE) method, which was used to select a set of features. RFE is a method of feature selection that removes the weakest features until the specified number of features is reached.

- normalizing the data and splitting the data in training and testing sets. The minimum and maximum values of all the numerical attributes should not have wide range of values. Feature normalization was performed by using `normalize ()` function. It can be performed by using `scale ()` function. The results of the normalization can be formatted in a data frame through `as.data.frame()` function.

- Data splitting involves partitioning the data into a training dataset used to construct the model and a testing dataset used to evaluate the performance of ML model later. In R language, `sample ()` function is used to sample the data. The most common splitting choice is to take 70 % of the original data set as the training set, while the 30% that remains will compose the test set.

- MICE package is used to impute missing values by using multiple techniques, based on the type of the data. Remove Outliers is performed by marking the outlier's values as N/A values. The value is considered as outlier if the value is greater than ($\mu + 3STD$) as mentioned in section 4.5 in page 77. Then all incomplete rows are Removed.

Each approach of this thesis is briefly explained in the following sub-sections. The importance of selecting datasets from the healthcare domain becomes apparent when addressing some health-care field challenges. Some countries face health-care challenges caused by inadequate medical staff and the shortage of modern rural hospitals, especially in rural areas [79]. Therefore, attempts have been performed to create various web-based medical diagnostic systems using different approaches to provide quick and straightforward access to diagnosis and medical advice such as rule-based mobile expert system [149].

3.4 Research methodology

The proposed framework is implemented and validated using three different approaches, i.e. 1) the applicability domain for the classification model approach, 2) the robustness of the classification model based on the applicability domain approach, 3) and the selection of a model based on the Pareto optimality approach.

Figure 50 in page 118 demonstrates the applicability domain for the classification model approach, the robustness of the classification model based on the applicability domain approach, and the selection of a model based on the Pareto optimality approach. In the sub-sections below, each method is explained shortly.

3.4.1 The applicability domain of the classification model

Machine learning algorithms rely on datasets fed into algorithms to execute the learning task. The ability to define the region of data space where a model can be used reliably is a necessary condition to ensure the robustness of the model outcomes on a new dataset. That can be a significant factor to determine the applicability domain of the model across data space. We focus on (reliable) and (not reliable) regions for building a classification model that has a reliable outcome. Consequently, we attempt to propose an approach based on the applicability domain (AD) plan to address the data locally. AD defines the extent to which a quantitative structure-activity relationship model (QSAR) can tolerate new compounds reliably [5][150].

Generally, any machine learning (ML) model needs to demonstrate not only good accuracy but also the reliability of the model. Many reviews and comparative studies on AD methods are available in the literature [151][119], which focus on distinguishing inliers from outliers, or high accuracy samples from low accuracy samples. In general, there is no global technique exists for identifying AD [152]. However, each AD definition usually depends on some arbitrarily defined distance to the training set instances for the given property. For reused models, the information should come about their applicability because of there a need to know how can reuse them. This research addresses this current problem in context reusing of the model. The estimating of AD using the ensemble approach is related to the concept of comparing the outcomes of several models constructed on different sets of data. So far, there is no clear focus on evaluating the classification model in term of pointing out for a new data point successfully (as may be adopted or not).

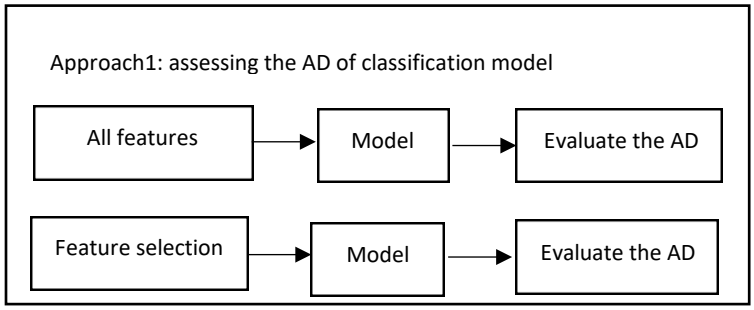


Figure 19: Various steps involved in the approach of assessing the AD of the classification model

Commonly the output of the ML algorithms only exhibits classifications for the new unseen dataset. Machine Learning algorithms provide an assessment of classifier's reliability based on an average performance on an independent dataset [152]. Several studies have done in reliability estimation performance of machine learning models.

Machine Learning algorithms have been effectively used in numerous classification issues over the previous decades [153][143]. While they generally outperform domain professionals considerably in terms of classification accuracy, they are mostly not used in practices [19]. The reason might be an unbiased assessment of the reliability of a single classification is difficult to achieve. A small study introduced by Kukar and Kononenko [154] proposed a method to estimate the classification's reliability. A comparison was performed on different domains and various Machine Learning algorithms. They offered a general transductive method for determining the reliability of classification on single examples, independent of the algorithm of applied machine learning. The fundamental concept of their study is to compare distinctions between inductive and transductive steps in the distribution of the probability of classification and use them to evaluate the reliability of single points (instances) in data space. This approach can represent its classifications as probability distributions . The assessment of the applicability domain of the classification model graphically presented in Figure 19.

3.4.1.1 The approach procedure

The approach process of the applicability domain of the classification model is described in this part briefly as follows:

a) The first stage of the implementation of this approach is to obtain some data sets.

b) Normalisation is applied to our dataset in which data values scaled into the range of [0, 1].

c) Next stage in the proposed algorithm is to divide the data into training and testing sets with an 70-30 split.

d) The following stage is to compute the Euclidean distance of the training set.

e) Compute the average of the distances between each instance and the remain instances from the training set.

f) Compute upper limit and lower limit.

g) Cov_i is calculated for each training examples considered as neighbours with distance values between the upper limit and the lower limit. The average of the neighbourhood width of all instances in the training set is denoted by AD_{all}.

h) Bootstrap aggregating (Bagging) generates a collection of new sets by creating samples from a given training set randomly with replacement. New classifiers are then trained for each sample of these new training sets.

i) Bias and precision are computed to measure the AD of the model. Ri is used to measure the reliability associated with each training example as explained in [115]. Figure 19 shows the approach process of assessing the applicability domain of the classification model.

3.4.1.2 The proposed algorithm

The concept of the density K-NN is used in this work to detect local neighbourhoods within the training data. The behaviours at the local level and the global level of any given dataset may be very different [115]. Therefore, a reference value is computed based on the list of average distances of each data point. This reference is used to assess the neighbourhood width for each single training example. The AD of the classifier will be defined based on the following steps: First, a Euclidean distance matrix of the training dataset is computed. This matrix will contain the distance sorted in ascending order of the distance between each training example and each of its training neighbours. Second, individual averages distances to the neighbours are calculated. This distance will be used around every training example as a neighbourhood width (coverage). Third, test new samples within the established coverage. If an instance is around any training instances within the coverage radius, it will be deemed to be covered by the AD. Establishing width addresses variability in data density throughout the dataset by setting distinct local the neighbourhood width (See Figure 26).

From the literature on identifying thresholds for distance-based approaches, no clear guidelines were apparent, and thus it is up to the user to define them [114]. In this research, thresholds defining strategies have been regarded for Euclidean distance measurement; the obtained findings have been compared. The Euclidean distance is much used and the most helpful distance measurement in QSAR research [114].

3.4.1.3 Feature selection

An optimal set of features used in the algorithm is created using the recursive elimination feature (RFE) method, as shown in the first chapter. The collection includes the top five features as well as the entire set of variables. The algorithm is tested for two sets of features. Recursive elimination of characteristics is a technique of selecting attributes used to remove the weakest features. Selection of

the features is a significant step, as the performance of the classification depends on the feature selection [46]. The framework of the approach of the applicability domain of classifiers is presented in Figure 20.

1 **Input** dataset $D = \{x_i, y_i\}_{i=1}^{n}$ n instances

2 Normalize the training set

3 Divide D into training set (D_{train}) and testing set (D_{test})

4 Compute $d_{ij} = dis(D_{train}, D_{train})$

5 Compute $d_{avreag} = \frac{\sum d_{ij}}{m}$, m is number of variables

6 Compute $\text{Upper limit} = \text{Q}_3 + 1.5 * \text{IQR}$, $\text{Lower limit} = \text{Q}_1 - 1.5 * \text{IQR}$ // for each element from d_{avreag}

7 For each sample in training set

Compute $d_{avreage} = \frac{\sum_{j=1}^{f} d_{ij}}{l}$

8 $f = \text{Lower limit} \geq dist\,(D_{train}, D_{train}) \leq \text{Upper limit}$, f is number of points which fall in distance equal or close to Upper limit.

9 End for

10 $Bootstrapresemble\,(k, D_{train})$ // Train resemble model with training set, k is number of models, prediction for each sample is obtained by each candidate.

11 Calculate STD and Agreement for each training sample based on the obtained predictions by candidates.

12 corresponding STD and Agreement to each sample.

Compute $Cov_i = d_{avreage} * (1 - STD) * Bias$

13 The coverage of the model equal to the mean of coverage of all samples. $AD_{all} = \frac{\sum_{i=1}^{n} Cov_i}{n}$, n is number of instances.

14 Check test samples in Cov_i and AD_{all}

15 **Return** AD_{all}

Algorithm 2: The applicability domainof classifiers.

This section presented the AD of a classification model (ADOC) proposed, which is inspired by the k-nearest neighbour approach. The AD for models derived using the KNN approach was computed from the similarities of distribution in the training sets between each compound and its nearest k neighbours [129].

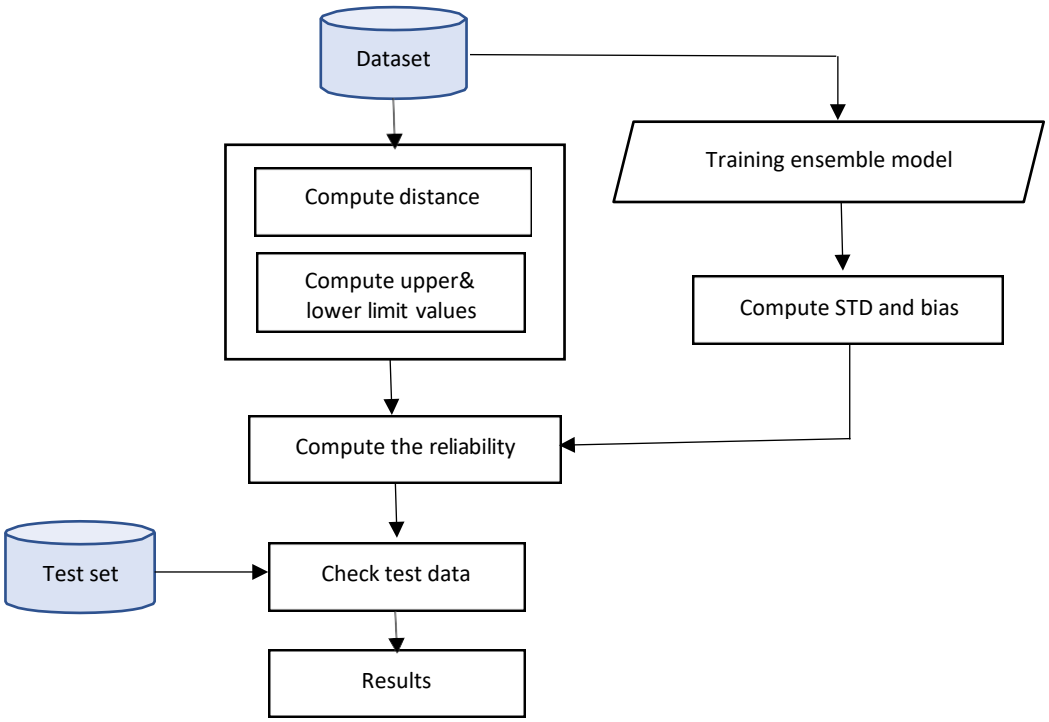


Figure 20: Overall methodology to estimating the applicability domain of the classification model approach

3.4.2 Robustness of classification model based on applicability domain approach

Robustness is a measure utilised in differing situations for machine learning models, for example, the capacity of the classifier to make the right predictions on the noisy dataset or a dataset with missing values. AUC measures have been accounted for improving the quality of robustness of classifiers in terms of measuring high sensitivity or true positive rate [25][27].

There are often normal or adversarial discrepancies between the learning sample and the environment of the evaluation. Thus, the classifiers should be robust and not sensitive to any changes in the distribution of the data. Some studies investigate methods for estimating the robustness of machine learning models [40][109]. Notably, state-of-the-art classifiers can tackle the classification problem with overall accuracy and with different input data such as RF [3] and SVM [6].

3.4.2.1 The proposed algorithm

This work proposes an approach to identifying the robustness of ML models depending on the AD approach. For several suggested QSAR research, KNN was the preferred choice [155][156].
Kernel Density Estimation (KDE) is a nonparametric technique for density estimation [115], and it is useful in detecting the highs and lows of point pattern densities (See Figure 21). It represents the distribution of data.
We applied the three phases of the model (See Figure 20) on given datasets, as is described in section 4.1. We consider RF algorithm [3] as a classifier. In machine learning, A random forest algorithm is a popular classification algorithm. it can be used for both classification and regression algorithm.
The random forest consists of decision trees. therefore, RF creates multiple decision trees and merges them to obtain a stable, accurate prediction and thus higher accuracy. A new synthetic data set was generated based on the original training dataset to validate the concept of AD for ML models. The proposed algorithm is implemented in three stages, defining the classification model, generating synthetic data points and evaluating the performance.

The first stage: Defining the classification model:
A model is built by applying a classification algorithm. Defining the classification model is to look for a classification algorithm that learns from a set of data. The process goes through several different steps such as pre-processing data, feature selection, dealing with missing values in the dataset and normalisation of the data. The RF algorithm is used for the classification of datasets.

The second stage: Generating synthetic data points.
The focus in generating the additional datasets is preserving the characteristics of the original dataset. We are motivated to this approach due to the effectiveness of the methods for generating synthetic datasets. For example, Multivariate Normal Distribution generator [157], MUNGE [157] and SMOTE [157][158]. The idea of STOME is determining the nearest neighbours of original data points to generate new synthetic samples for balancing imbalanced classes. MUNGE is a strategy to create synthetic data for model compression. For the MUNGE method, a large model can be replaced with a smaller model which can mimic its behaviour efficiently. Despite, both the training set and test set have similar statistical characteristics, and only the training set is used in creating synthetic points.

The third stage: Evaluating the performance.

Make predictions after applying the model, and different points will be considered to reflect upon the obtained results. The description of the AD of a model in the space can be used to define the robustness in the results derived. The predictive ability of the model reflected by using the accuracy and classification error. Since the data points that have poor accuracy are considered as unreliable predicted (outside the domain of the model). The model was evaluated as the following parameters:

- Rate of samples that have good accuracy.
- Rate of examples that have poor accuracy.

The robustness of the model is evaluated based on the prediction of the model or the ability of prediction. Figure 17 demonstrates a schematic representation of density and reliability mapped

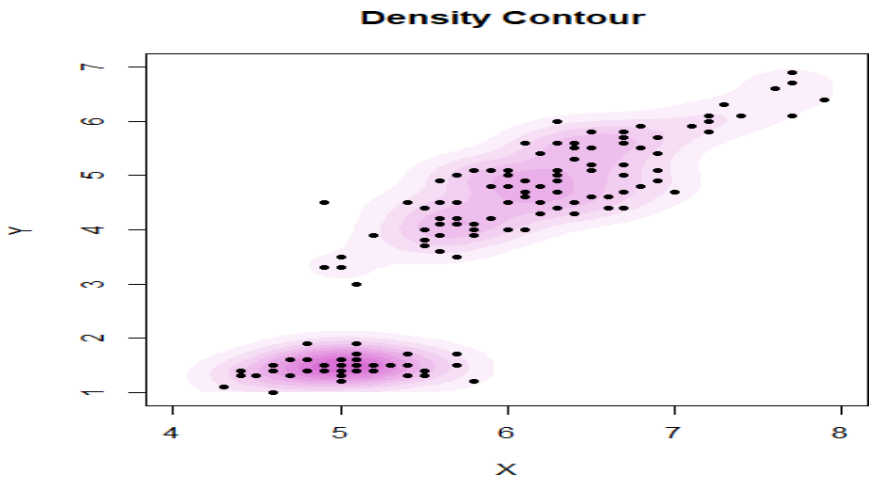


Figure 21: Two variables density estimation where the data fall

across data space displaying densely populated and more reliable areas in a darker region (highly dense region), transitioning sparse and unreliable data into white regions (lower dense region).

Kernel density estimation for a set of data can be applied and decide whether the point lies in the space compared with other areas. It includes bandwidth determination, defining the levels and the corresponding colours [159].

By obtaining the highest value (the maximum value), and the lowest value (the minimum value), a more detailed picture of the data can be obtained. The five-number summary is included the minimum amount, first quartile, median, fifth quartile and maximum value.

The function `dist()` for computing the distances between observations in two matrices and return a matrix is described in [160]. The framework of the approach of the robustness of the classifiers based on applicability domain approach is presented in Figure 22.

Table 12: Summary of notations in the algorithm 3

symbol	Description
D	A given data set
r	Threshold, Positive number between 0 and 1
Sub_{indise}	Dataset that are in the model domain (correct predictions by the model
$Sub_{outside}$	Dataset that are out the model domain (incorrect predictions by the model)
D_{test}	Testing data set
D_{train}	Training data set
C_i	A built classifier
Sub_t	Data set obtained by adding threshold r to chosen points from test data set (the points between t and max distance)
Su_n	Data set that fall in Sub_t interval
d	A distances matrix of the train data set
$MAXdist$	Maximum value in distance matrix(ds)
t	A value of distances; Where, t<= $MAXdis\ (ds)$
N	Numberof iterations adding r
$\hat{y}$	Predicted class label
y	The class of a point
sd_s	Numberof rows in $\mathrm{Sub_n}$
nr	$\mathrm{umber\ of\ row\ of\ Sub_n}$
ma	Maximum value of distance

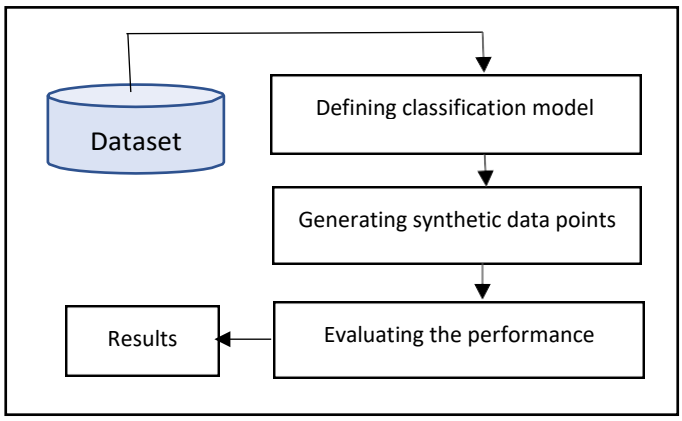


Figure 22: Overall methodology to estimating the robustness of the classification model approach

Algorithm 3 Evaluation the algorithm

Input: Dataset D

Threshold r,

Base learn classifier C

Number of iteration N

Output: Two subsets Sub_{inside} , $Sub_{outside}$

(1) Normalize D, and Remove Outliers from D

(2) Divide D into training set D_{train} and testing set D_{test}

(3) Build classifier $C = Beaslearn(D_{train})$

(4) Evaluate the classifier *C*

(6) $d = \{dist\ (x_i, x_j): x_i, x_j \in D_{train}\}$

(7) $d = \max\{\delta(x_i, x_j): x_i, x_j \in D_{train})\}$

(8) Ref= Q3+(1.5*IQR)

(9) t= mean (subset (distance1, distance1>=Ref))

(10) $Sub_t = subSet(D_{train}, d, t)$

(11) Input r, which 0≤r≤1

(12) Repeat

(13) Add r to each element in Sub_t; where $0 \le r \le 1$

(14) $sd_s = dist(D_{train}, Sub_t)$

(15) $\mathrm{Sub_n} = \mathrm{subset}(\mathrm{D_{test}}, \min(\mathrm{Sub_t}) \le \mathrm{D_{test}} \le \max(\mathrm{Sub_t}))$

(16) $\hat{y} = \mathrm{C}(\mathrm{Sub_n}),\ \mathrm{i} = 1, \ldots, \mathrm{nr}$ // C is a classifier

(17) For all samples in $\mathrm{Sub_n}$ do

(18) if ($y_j = \hat{y}$) do

(19) add $\mathrm{Sub^i_n}\ (x_j, y_{j)}$ to $\mathrm{Sub_{inside}}$

(20) Else

(21) add $\mathrm{Sub^i_n}\ (x_j, y_{j)}$ to $\mathrm{Sub_{outside}}$

(22) End if

(23) End for

(24) Keep $\mathrm{Sub^i_{inside}}$, $\mathrm{Sub^i_{outside}}$

(25) N=N-1

(26) Until N=0

(27) **Return** $\mathrm{Sub^i_{inside}}$, and $\mathrm{Sub^i_{outside}}$

Algorithm 3: The robustness based on applicability domain approach algorithm

3.4.3 Defining the optimum classifier

After getting the ensemble classifier from the approach of assessing the AD for classifiers in Chapter 4, the available collections of models are used for finding a better model among them. The identification of this model is performed based on Pareto optimality, which, mines model collections and identifies a model that offers a good performance on the test set.

3.4.3.1 The model

There is a set of m classifiers $M = M1, \dots, Mm$ associated with class y. These obtained models (or classifiers) have different performance. To identify the best model from the collection of models M for a data test x, we create a model to select a classifier with a maximum average of neighbourhood width and maximum accuracy. The framework for defining the optimal classifier is presented in Figure 19. It gives the procedure of the approach. A set of models built based on s set of data are performed differently.

We aim to investigate if these models can perform better for data that lie in the AD of the model. In this section, we introduce an approach to define a reliable classifier from a collection of existing

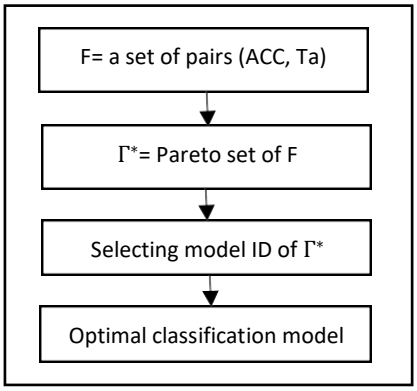


Figure 23: Overall methodology

classifiers for test data. Consider X is a set of data points with features, and there is a set of models M. For each $x \in X$, classifier outcomes, $\acute{y} = \acute{y_1}, \dots, \acute{y_m}$ for models M are known. For defining a model for a given dataset, we make a set of pairs (T, ACC), where ACC denotes the average of the accuracy of the model on the test set. The threshold T represents the average of thresholds of the model on training set instances. ACC defines the performance for each classifier. From these pairs, we can find models that have the maximum accuracy and maximum threshold. This can be found in the topic of multi-objective optimisation. Across all conditions, no solution exceeds the others. Consequently, we

have a set of solutions that cannot be compared with each other. Instead of finding one solution, the Pareto set can balance results provided by available models with accuracy and reliability.

Many real-world issues require many objectives functions to be optimised simultaneously. Some of these objectives may conflict with each other. For instance, finding an ideal classification model of relevant models from available collections of classifiers, where the goals may include minimising the computing costs, minimising the error rate, minimising the time, and maximising the accuracy performed. The MOO [161] issue addresses a finite number of (goals) objectives functions. For an optimisation issue with n equal importance objectives, the minimisation (or maximisation) of all the objectives is required to deliver a performance criterion. In general, MOO can be done by the following steps:

- Applying the constraints.
- Finding the feasible solutions.
- Obtaining the optimal solutions that satisfy all the constraints.

Different solutions are represented in the feasible solution space (Figure 24), x is defined in 2D space as $x = (x_1, x_2)$. It is called decision variables space.

Machine learning classification algorithms are very well suited for dealing with the optimisation of multiple variables, the concept of Pareto optimality can be useful when simultaneously optimising multiple objectives [162]. This section provides a brief overview of multi-objectives optimization (MOO), focusing on definitions that are needed in later part.

3.4.3.2 Pareto optimality

The concept of Pareto-optimality was introduced at first time by the mathematician Vilfredo Pareto. Pareto optimality is a concept built on multi-objective optimisation that promotes multi-objective vector optimisation by trade-offs between multiple-objectives combinations [106], [163]. The trade-off is constructed to enhance the performance of a goal at the cost of one or more other goals [164]. As shown in Figure 20 each point in the objective space represents a unique set of model features, so Pareto optimality classifies multiple Pareto points (solutions) [33].

This section provides a few definitions that are needed when talking about MOO. Definitions include dominance, pareto-optimal, and pareto-optimal front. These definitions assume minimisation.

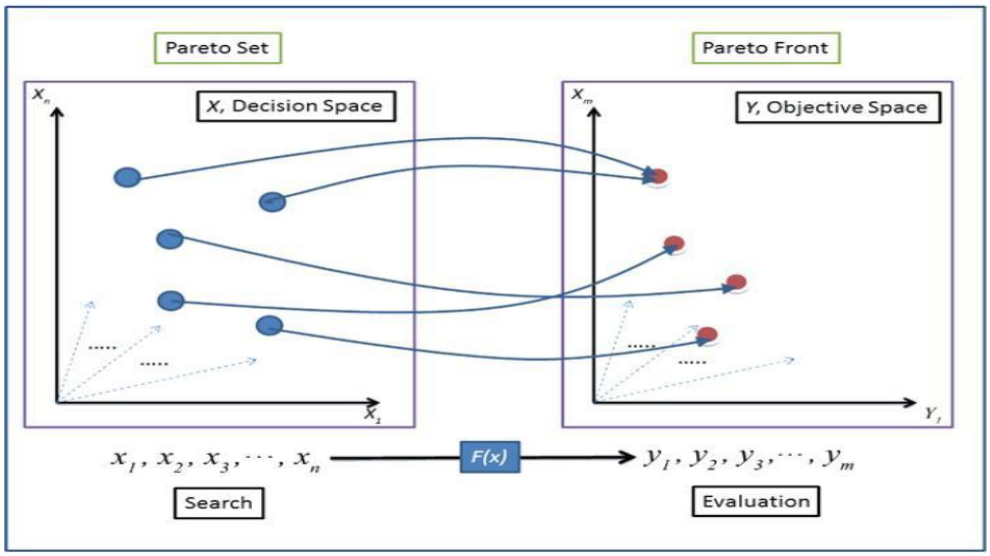


Figure 24: Multi-objective optimization problem: mapping the search space to the objective space [19]

3.4.3.3 Pareto points and their properties

Let $\mathcal{S} \subseteq \mathbb{R}^n$ denote n-dimensional space, and $\mathcal{F} \subseteq \mathcal{S}$ the feasible space. consider a decision vector

$x = (x1, x2, \dots, xn) \in \mathcal{S}$, a single objective function, $f_j(x)$, is identified as $f_j : \mathbb{R}^n \longrightarrow \mathbb{R}$. Let $F(x) = f_1(x), \dots, f_k(x)) \in \mathcal{O} \subseteq \mathbb{R}^n$ be an objective vector. The objective space is denoted as $\mathcal{O}$, the decision space is $\mathcal{S}$.

Definition 1 Domination: a decision vector, $x1$ dominates a decision vector, $x2$ (denoted by $x1 \prec x2$), if and only if

- $x1$ is not worse than $x2$ in all objectives, $i.e. f_j(x1) \leq f_j(x2), \forall j = 1, \dots, k,$
- $x1$ is strictly better than $x2$ in at least one objective, $i.e. \exists j = 1, \dots k : f_j(x1) < f_j(x2).$

Definition 2 Weak domination: a decision vector, $x1$, weakly dominates a decision vector, $x2$ (denoted by $x1 \preccurlyeq x2$), if and only if

- x is not worse than y in all objectives, $i.e. f_j(x1) \leq f_j(x2), \forall j = 1, \dots, k,$

Definition 3 pareto-optimal: a decision vector, $x^* \in \mathcal{F}$ is Pareto-optimal if there does not exist a decision vector, $x \neq x^* \in \mathcal{F}$ that dominates it. That is, $\nexists: f_j(x) < f_j(x^*)$. An objective vector, $f^*(x)$, is Pareto-optimal if x is Pareto-optimal.

Definition 3 pareto-optimal set: the set of all Pareto-optimal decision vectors from the Pareto-optimal set, Γ^* . That is,

$$\Gamma^* = \{x^* \in \mathcal{F} \setminus \nexists x \in \mathcal{F} : x \prec x^* \} \quad (22)$$

The Pareto-optimal set therefore contains the set of solutions, or balanced trade-offs, for the multi-objective problem (MOP).

Definition 2 pareto-optimal front: Given the objective vector, $F(x)$, and the Pareto-optimal solution set, Γ^* , then the Pareto-optimal front, $\Gamma\mathcal{F}^* \subseteq \mathcal{O}$, is defined as

$$\Gamma\mathcal{F}^* = \{F = (f_1(x^*), f_2(x^*), \dots, f_j(x^*)) | x^* \in \Gamma \} \quad (23)$$

The Pareto front therefore contains all the objective vectors corresponding to decision vectors that are not dominated by any other decision vector [163]. The concept of dominance is illustrated in Figure 21 [165], for two-objective function, $F(x) = (f_1(x), f_2(x))$. The striped area denotes the area of objective vectors dominated by F.

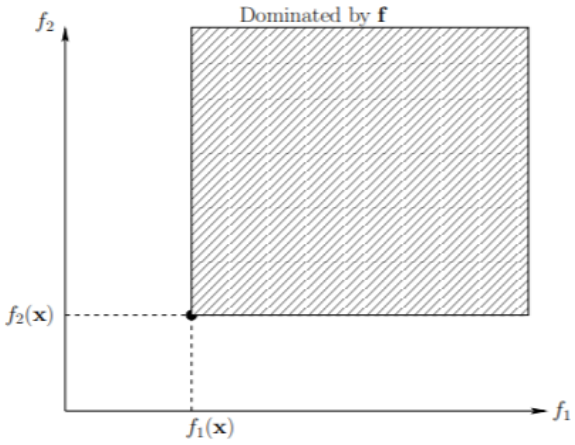


Figure 25: Illustration of dominance [151]

```
    Algorithm 4: FindingParetoset(P)
1    Let P = {A1, … , An} be solutions
2    Let P0 := {∅}
3    Let i = 1, j = 1
4      For each combination C ∈ P {
5        let C ∈ P0, If Value(Ai) < Value(Aj) }
6    Repeat from step 2 until no more C is added to P0.
7    If i = n, stop. Otherwise i = i + 1 and go to step 5.
```

Algorithm 3: Finding Pareto set

The above definitions and basic properties of Pareto set can be found in [161][165][166]. The next section provides the proposed model procedure.

3.4.3.4 The procedure of the model development

The procedure of the model development is as follows:

1) Obtain test and train set.
2) Create a set of pairs $F = (ACC, T)$.
3) Find Pareto set Γ^* for F .
4) Choose the most appropriate model for the test set.

3.5 Summary

The objective of this chapter is to describe the dataset and the research methodology. One of the important parts of research is the data. Along with its limitations, the data description is provided. Although the data in this database represent an important contribution to knowledge about the AD of the classification model, there are limitations of the data which must be recognized. The datasets samples are small, errors in the data and some datasets are imbalanced which cannot reflect the situation of its target patients well.

The research methodology is presented in this section with three methods, i.e. The applicability domain of the classification model, Robustness of classification model based on applicability domain approach, and Selecting optimum classification performance model. The performance criteria are also provided to assess the efficiency of each used approach. The results of each approach are provided in Chapter 4, 5 and Chapter 6.

4 The applicability domain of classification models

4.1 Introduction

There are approaches of assessing the classifiers based on AD methods such as the STD method [85] and the k-nearest neighbours' density (dk-NN) [143]. These techniques map the outcomes of new examples in term of distance to the model space while considering the reliability of nearby training instances. Accordingly, here, we used a reliability measure that results from two distinct effects, bias and precision [115] for the classification model. This section discusses some aspects that influence the performance of classification models. This work includes the proposed algorithm, The AD of the classification model, and feature selection of AD.

4.2 The procedure of the proposed algorithm

The applicability domain of the QSAR model characterises the restriction of the model in its structural area and reaction space [152]. The model validation process can restrict the applicability of a model to predict reliably new samples that have a similar structure to the training samples [77]. We are motivated to look closely at possible solutions inspired by the density k-NN (dk-NN) approach proposed by Sahigara et al. [143] and the reliability of the predictions of a QSAR models mapping approach presented by Aniceto et al. [115]. We have employed the algorithm discussed in (Algorithm 2) above, to assess the AD of the classification model. In the following section, we recall the stages of the algorithm implementation.

The first stage of the implementation of this approach is to obtain some data sets, as shown in Table 9. Each dataset $D = \{x_i, y_i\}_{i=1}^{n}$, containing n points in d-dimensional space, which $x_i \in R^d$ with corresponding y_i. Let $D = D_1, \ldots, D_k$ be input dataset. K random samples with replacement from D. Let $\{c_1, c_2, \ldots c_m\}$ denote the set of m class labels. Let M be a classifier. Let $\bar{y} = M(x_i)$ denote its predicted class for a given example x_i, $x_i \in D$. Where, $x = (x_1, x_2, \ldots, x_d) \in R^d$. x is a point in d-dimensional space. Normalisation is applied to our dataset in which data values scaled into the range of [0, 1] [28]. Data needs to be normalised because the study relies on the Euclidean distance measure [167].

Next step in the proposed algorithm is to divide the data into training and testing sets with an 70-30 split [102]. The training set is denoted by (D_{train}) to train the model while the testing set is denoted by (D_{test}) to test the trained model [168]. The following stage is to compute the Euclidean distance [28] of the training set. The distance can be used as a metric that identifies the similarity for a specified property of a model between the training set examples, and the testing set samples [1][152][92]. From the Euclidean norm, we can define the Euclidean distance between x and y data points the training samples, as follows:

$$d_{ij} = \sqrt{\sum_{i=1}^{m}(x_i - y_i)^2} \tag{24}$$

Where, d_{ij} is the distances between the examples in the training set, and m is number of features. The square root of the sum of squared differences between the vector components of the two variables is the Euclidean distance.

Then the average distance for each instance in training set is computed as the average of distances between this instance and the remain instances from the training set. According to Tetko et al used Euclidean distances measure to define the similarities between a molecule i and a training set. The AD for models derived using the KNN approach was computed from the similarities of distribution in the training sets between each compound and its nearest k neighbours. The average Euclidean distance to the nearest k neighbours of this molecule in the training set was calculated using only a subset of variables identified as optimal by the modelling procedure. The distance distribution in the training set between each compound and its nearest k neighbours is calculated to produce an applicability domain threshold, which calculated for each KNN model [92].

The average distance for each instance is denoted by $d_{avreage}$, it is defined by follows:

$$d_{avreage} = \frac{\sum d_{ij}}{m} \tag{25}$$

Where, $d_{ij}$is the distance of a data point to the rest of training data points, and m is number of the data points in the training set. Compute the coverage threshold corresponding to instances in the training set.

Then these averages $d_{avreage}$ are used to compute a reference value (upper limit). The upper limit value is used to select data points that are allocated in the coverage of any point, as shown in Figure 22. Each data point has some data points in the upper limit value range. Assume the grey coloured point is a point i from the training set.

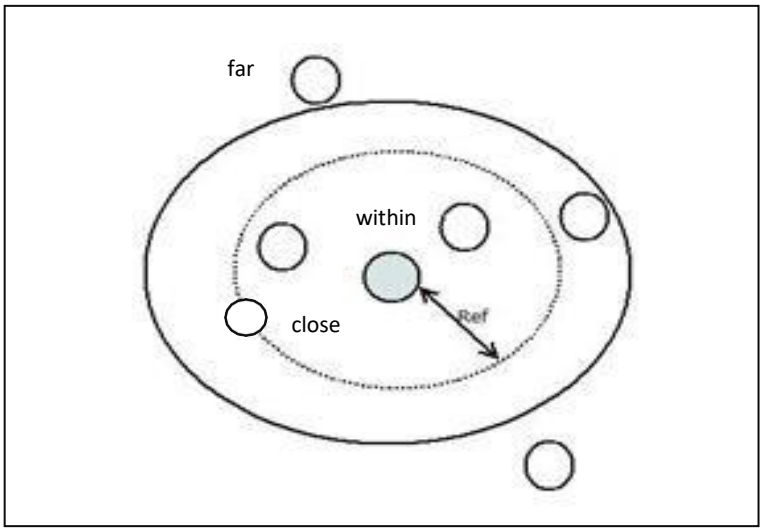


Figure 26:The neighbourhood width of data point

There are two testing instances are covered by the training instance i, and one test instance is close to the training instance. The rest of the points are outside the coverage of this data point. The reliability of the distance depends on upper limit value attributed to training instance i. upper limit value is computed based on Tukey's fence method [25] by using the following equation:

$$Ref = Q3 + 1.5 \times IQR \quad (26)$$

Where, Q3 denotes the 3rd quartile, and IQR denotes the interquartile range.

One of the most common variation measurements used in statistics is the interquartile range [161]. It is a measure of how the mean spreads information and breaks the data set into quarters. It is calculated as the difference between the 3rd and the 1st quartiles. The fundamental formula is:

$$\mathrm{IQR} = Q3 - \mathrm{Q1} \quad (27)$$

Table 13: Summary statistics of training set from Pima dataset

Characteristic	Mean	Standard Deviation	Median	Q1	Q3	IQR
The average of distances	0.59961	0.11961	0.564	0.517	0.64775	0.13075

Quartiles give a full picture of the data set. The first and third quartiles provide information about the internal data structure. Between the first and third quartiles, the middle half of the data is centred around the media. The difference between the first and third quartiles shows how the median data is arranged. A small range of interquartile indicates data clumped over the median. A wider interquartile range means more dissemination of data [169].

Table 14: Outliers in characteristics measured of training set from the datasets (The average of distances is considered)

N	Data set	Lower Limit1	Upper Limit2
1	Pima dataset	0.321	0.844
2	Breast-cancer dataset	0.659	0.973
3	Indian liver patient data	0.390	0.942
4	Heart dataset	0.435	0.930
5	Thyroid dataset	0.409	0.951
6	Cardiotocographic dataset	0.537	0.921
7	Hepatitis	0.207	0.911

We illustrate Tukey's fence method using all used dataset [145]. The summary statistical of training set for the dataset are summarised in Table 13 and Table 14, the median of the average distance is 0.564, and the quartiles are determined as the lower and upper halves, respectively. The first quartile (Q1) is the mean of the two middle values in the lower half. The same way is used in the upper half to determine the third quartile Q3=0.648. The interquartile range is defined as (IQR) where IQR= Q3-Q1= 0.13075. Outliers are values below the lower limit (Q1-1.5(Q3-Q1)) or the upper limit (Q3+1.5(Q3-Q1)). The Euclidean distances between each training instance and all training examples is calculated to obtain Cov_i. The average distances Cov_i are calculated for each training examples considered as neighbours with distance values closer or equal to the upper limit, which, $Cov_i = \{ d_{ij}, lower\ limit \leq d_{ij} \leq upper\ limit \}$. From Figure 27 represents the average distances to the instances with distance value closer or equal to upper limit value for Brest cancer dataset [159]. From Figure 27 most values of the average distances for Brest cancer dataset have about 0.8 ofaverage.

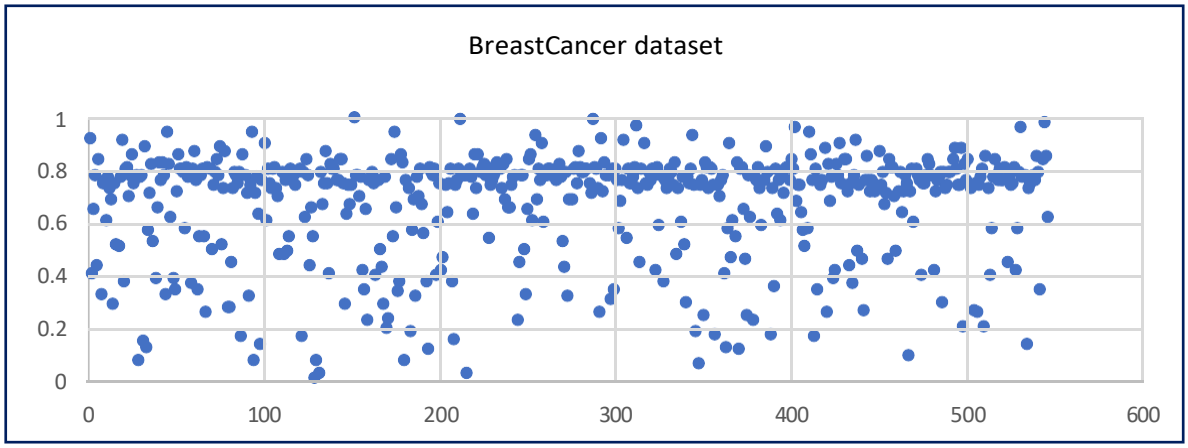


Figure 27: The average of the distances for all points that are equal to upper limit value on Breast Cancer dataset

Cov_i is the neighbourhood width threshold of each point. The distances between each training example and the rest of the cases are calculated to generate the AD threshold for each instance. The variation of data density across the dataset is addressed by computing different local thresholds for each sample. The neighbourhood width is based on their reliability around each training example.

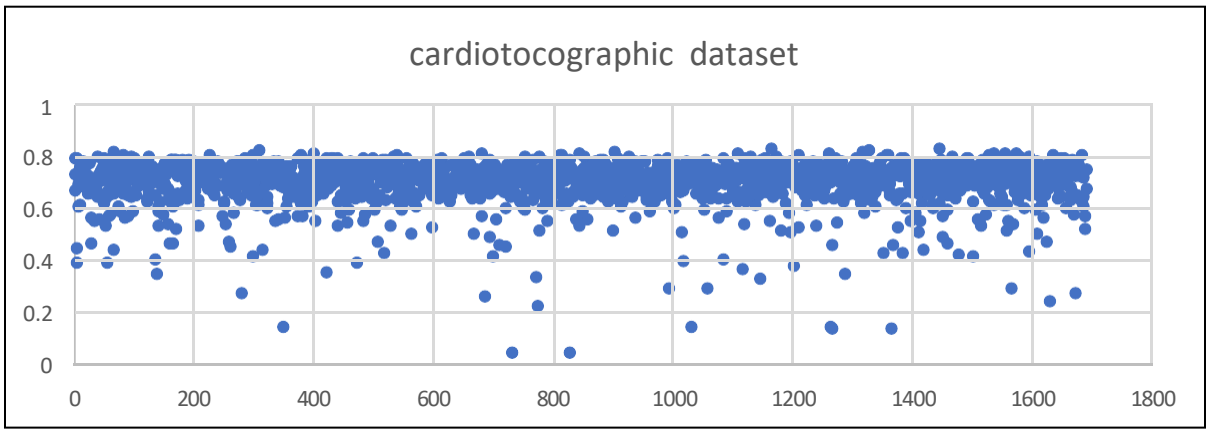


Figure 28: The average of the distances for all points that are equal to upper limit value on cardiotocographic dataset

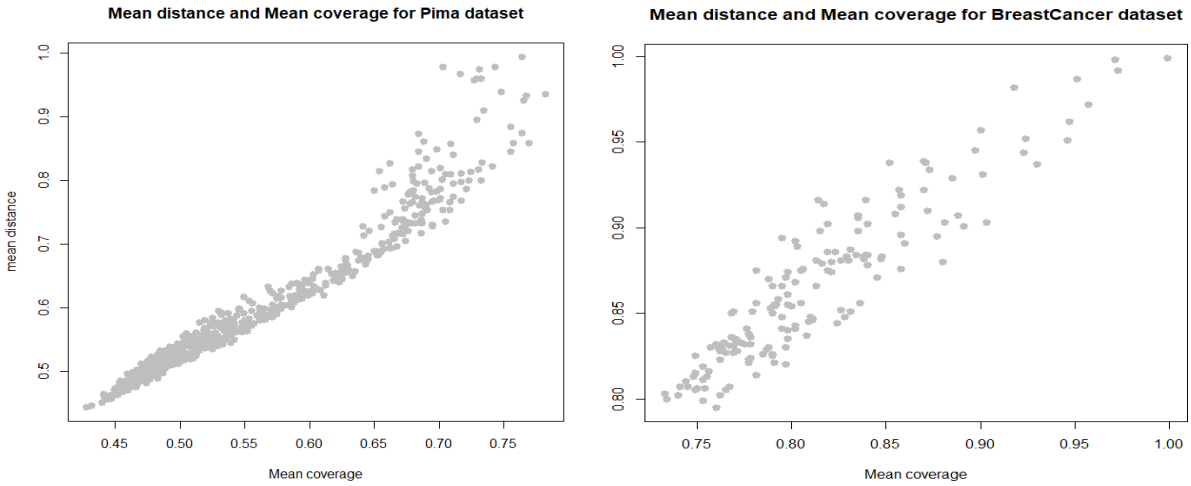
From Figure 28 most values of the average distances for cardiotocographic dataset have value between 0.6 and 0.8 of average.

Figure 29 shows the relationship between Cov_i and the average distances of each instance from the rest of the data for Breast Cancer dataset and Pima dataset. Mean coverage denotes the average of distances within upper limit value for each data point (it is also called the neighbourhood width). Mean distance indicates the average distances of each instance from the rest of the training set.

The average of the neighbourhood width to all instances of the training set is denoted by AD_{all} (See line 12 in Algorithm 2). It can be computed as follows:

$$AD_{all} = \frac{\sum_{i=1}^{n} Cov_i}{n} \quad (28)$$

Where, n is number of examples. The data spread differently regarding the degree of occupying the regions [92][115]. The density k-NN (dk-NN) of AD approach [4] works based on the average overall distance to the k-the nearest neighbour, which may cause some instances may have no neighbours. So, we used both bias and precision to measure the AD of the model as clarified underneath. Figure 29 represents the spread of values in the data by using the mean. A large spread indicates that there are probably significant differences between individual values of the data. Additionally, a small variation in each data group indicates that the similarity of the data values.



(a) Pima dataset (b) Brest Cancer dataset

Figure 29: Relationship between mean overall distance and mean neighbourhood width distance for (a) Pima dataset, and (b)Breast Cancer dataset.

4.3 Bias and precision for assessing the applicability domain

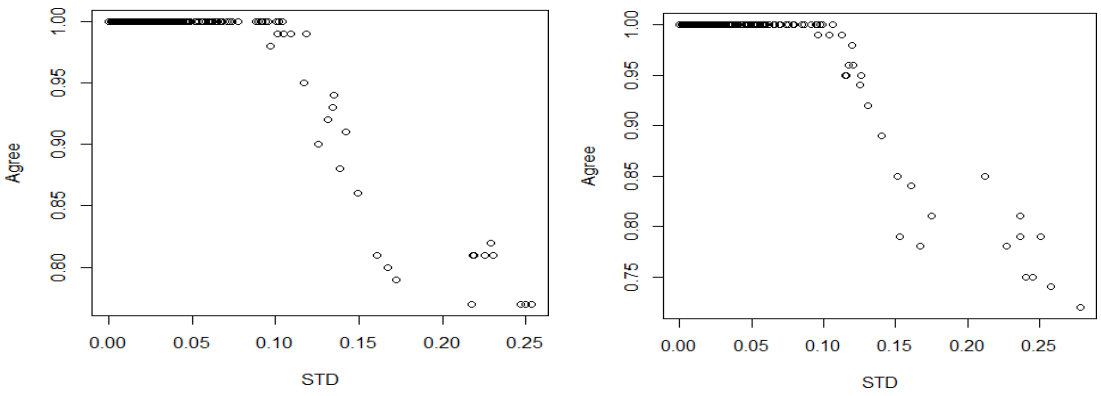
The bias of a classifier represents the systematic deviation between its predicted decision boundary and the real decision boundary [28][170], whereas the precision of a classifier C for class ci is defined as the fraction of correctly classified points overall points predicted to be in class ci, $precision_i = \frac{n_{ii}}{m_i}$, Where, m_i denotes the predicted examples as ci by classifier C [28].

According to Aniceto et al combining some measures such as the bias and the precision might be an appropriate correction factor for the applicability domain assessment of the model, which estimates the reliability associated with the model [115]. The coverage around each instance of the data is computed according to their reliability as a composed of precision and bias terms.

The term bias was introduced in 1980 by Tom Mitchell in [171][172]. The idea behind introducing this term is based on the ML model may give importance to some features of a larger dataset for better generalisation. Therefore, the obtaining model can be less sensitive to some data points [172]. The problem can be worse when the algorithm biased on some features even after omitting fewer essential features. The challenge is raising concerns as ML models have started to play roles in various crucial decisions such as medical diagnosis. Therefore, the bias in machine learning would affect the result of the considered ML models. For example, Amazon attempted to build a recruiting system on the ML model, which ended up being biased against women [108]. The system showed unexpected behaviour. It tended to reject applicants based on their gender because the model trained on datasets mostly came from men. Thus, this biased behaviour led to reconsider the system by the company and lose full trust in the rankings provided by the model.

In this algorithm, we make use of an ensemble learning [96] that combines many models M to improve the outcomes of the classification models as we explained in Chapter 2. The ensemble methods make a combined classifier using the result of multiple base classifiers. Each learner or classifier differs in its decision, so the learners complement each other. Combining the learners can be achieved based on different learners or depends on a different subset of the training dataset [15][13]. The standard deviation (STD) of predictions of models are correlated with the precision (accuracy) of the outcomes [173][174]. Therefore, the standard deviation of model results is considered as a metric characterising the distance of molecules from the ensemble of models [152]. Obtaining a high level of agreement and thus, a smaller standard deviation (STD) means gathering more reliable predictions. Stable performance is found at that expect regions where can generate more robust predictions.

Moreover, predictions are less susceptible to changes in the data partition within the ensemble within the learning task (See Figure 25). Alternatively, regions with a less stable will rely significantly on the data partition used, thus generating more considerable differences between different models. The neighbourhood width around each instance of the data is computed according to their reliability as precision and bias.



(a) being class (b) malignant class

Figure 30: Relationship between agreement and ensemble standard deviation in the Breast Cancer dataset, (a) being class, and (b) malignant class

4.4 Building ensemble classifiers

Some of the classification algorithms can reuse one or more current classification algorithm by either applying multiple models for robustness or by combining the outcomes of the same algorithm with distinct components of the data. Ensemble learning methods have been used in data mining and artificial intelligence extensively [175][176][177]. A previous study has reviewed ensemble learning techniques and explained the ability of ensembles for providing better performance than a single classifier [24]. Bootstrap aggregating (Bagging) generates a collection of new sets by creating samples from a given training set randomly with replacement. New classifiers are then trained for each example of these new training sets. They are combined in different ways, such as a majority vote [26][95].

Bootstrap aggregation is an ensemble classification method which use various bootstrap samples (with replacement) from input training data D to produce slightly distinct training sets D_i. Base classifiers M_i are learned on D_i. An ensemble classification method of bootstrap aggregation uses multiple bootstrap samples with replacement from the training set for generating different training sets. Each training set is used to learn a classifier [173][175]. The C base classifiers M_i are used to classify a given test point x. Let the number of classifiers that predict the class of x as c_j be given as follows:

$$p_j(x) = |\{ M_i(x)\ c_j / i = 1, \dots, C \}| \quad (29)$$

The combined classifier is denoted by M_C, it is used to predict the class of a test point x by majority voting among the classes as:

$$M_C(x) = argmax\{ p_j(x) / j = 1, \dots, C \} \quad (30)$$

Or by sign (summation of classifiers predictions) as:

$$M_{C(x)} = sign\ (\sum_{i=1}^{C} M_i(x)\ ) \quad (31)$$

When the base classifier is unstable such as decision tree, the bootstrap can reduce the variance and the bias [170].

```
Bootstrap resampling (k, D) // D is the data set
    For i=1 to k do
       Di = sample of size n from D,i=1,…,k
       Mi=train classifier on Di
       θi = assess Mi on D
    End for
μθ = (1/k) Σ_{i=1}^{k} θi
δ² = (1/k) (θ_i − μ)²_θ, δ = √δ²
Return μ_θ and δ²
```

Algorithm 4: Bootstrap Resampling Method

Algorithm 4 shows the pseudo-code of bootstrap resampling method. Estimating AD using ensemble approach related to the concept of comparing the outcomes of several models constructed on different sets of data [14].

4.5 The applicability domain of the classification model

The neighbourhood width [115] around each training instance is estimated according to their reliabilities, which is measured by using bias and precision. High variance classifier is unstable and tends to overfit the data and has poor generalisation performance [115][178]. The overfitting happens when the tree grows full until all leaves are pure and the training error equal to zero. The model learns from the training data, including the noise. For instance, decision trees classifier is subject to overfitting training data. This issue can be solved by pruning a tree after learning to extract some of the noise [15].

On the other hand, high bias leads to underfit the data. Underfitting refers to a model that cannot generalise the training data to the new dataset. Consequently, poor performance on the training data

training will be obtained by the model [170]. The solution can be by using alternative algorithms for machine learning. Generally, learning aims to decrease classification error by reducing bias or variance. There will be a higher variance in smaller training data sets. Various forms of analysing the ensemble try to minimise this bias and variance. For a discussion on bias and variance, the reader is referred to [28][95].

For obtaining reliable predictions, the ensemble models would have a high degree of harmony and thus a smaller standard deviation (STD). Mathematically, sample standard deviation (STD) can be computed as:

$$STD = \sqrt{\sigma^2} = \sqrt{\frac{\sum(y_{mi} - \mu)^2}{n-1}} \tag{32}$$

Where, σ^2 is the variance.

The standard deviation is a widely used measure to summarise the amount of dataset about the mean where the most data lie. It aids to determine the proportion of the values which lie rang of the value of the mean for the dataset that has a normal distribution. 68% of values are less than ±1sd away from the mean, 95% of values are less than ±2sd away from the mean, and 99% of values are less than ±3sd away from the mean [179][168].

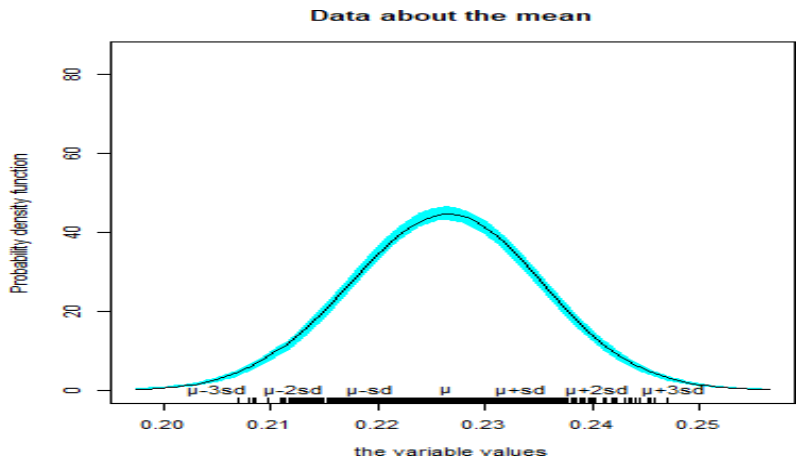


Figure 31: Idealised normal distribution showing area corresponding to 1,2 and 3 standard Deviation (STD)

The STD represents the data falls within plus or mince STD around the mean [180], as shown in figure 26. The mean visualises the centre of the data and STD tells us how aboard the data around the mean. Vary large of STD means lots of values lie on both sides of the mean. Very small STD shows very tightly compact data set around the mean [181].

Theoretically, based on the principle of an ensemble (set) of models, M_C (defined in section 4.4) will have a high level of agreement and thus, M will have a lower standard deviation (STD) for more reliable

outcomes. It would be expected that areas, where robust outcomes are yielded that, are less subject to the change in the learning process, including changing the subset within the ensemble learning method. Otherwise, areas with less stability will depend heavily on the used subset, therefore, creating broader distinctions between distinct models [115]. However, STD values assess the level of precision only. It is necessary to use the rate of agreement between the set of outcomes and the actual responses to overcome cases of systematic bias towards an incorrect classification. Precision implies that measurement using a specific tool or implement generates comparable outcomes each time it is used. For instance, if a scale is used for five times in a row, each time the same weight will be obtained by using this accurate scale. Mathematically, calculating precision is essential to determine whether used tools or measurements work well enough to gain useful results. The precision of any dataset can be recorded using the range of values, the average deviation, or the standard deviation [168]. A systematic bias occurs when most wrong outcomes are close to each other. The algorithm can capture these outcomes as results of high reliability when only a correction of STD was used. As a result, the combination of bias and precision is a suitable reliability measurement factor, Ri.

The term (1-STD) is used to measure precision for each instance in training set.

$$\mathrm{Ri} = (1 - STD) * Bias \tag{33}$$

For each training instance, the with coverage Cov_i. the precision measure can be calculated as:

$$Precision = 1 - STD \tag{34}$$

The second factor of reliability measurement is the bias measure which represents the agreement. The agreement refers to the degree of bias in a set of outcomes [28]. The agreement is calculated from the amount of matching observed and predicted responses in an ensemble of models, divided by the total number of models in the ensemble [152][115]. The ensemble standard deviation is calculated according to:

$$Bias = \frac{y_i - \hat{y}}{M} \tag{35}$$

Where, y_i denotes the expected output, y_i is the actual value and M is number of classifiers in the ensemble [172]. Ri is used to measure the reliability associated to each training example [182][115]. From equation 21,22, and 23, we can rewrite wi as follows:

$$wi = (1 - \sqrt{\frac{\sum(y_{mi} - \mu)^2}{n - 1}}) * \frac{y_i - \bar{y}}{M} \tag{36}$$

Where y_{mi} denotes the predicted class for the example i by the mode m within M model. μ is the average prediction outcome by M. Figure 25 shows the relationship between agreement (Bias) and ensemble standard deviation in the Breast Cancer dataset, (a) being class, and (b) malignant class. As we mentioned previously STD is the standard deviation of outcomes obtained from the ensemble of models, Thus, 1 − STD is the precision rate of these outcomes. For each instance from training set, w_i will be used to calculate the AD of each point.

4.6 Evaluation of the proposed algorithm

In order to evaluate the performance of the current approach. datasets from UCI repository [145] were used. The description of the datasets is found in section 3.2. decision trees algorithm is used to build the classifier. Decision Trees are commonly used in machine learning for classification and regression tasks [87]. It has influenced a wide area of machine learning, which can be used to visually and explicitly represent decisions [84]. DT improves continuously, for example, CART's algorithm and pruning ideas [19], [183]. DT learning algorithms have a long history and theory on how to split the data. But will obtain model work well on new data as for training data considering the AD concept of the model. The training was done using the R package. The optimisation was done with respect to Recursive feature elimination (RFE) method, which was used to select a set of features. RFE is a method of feature selection that removes the weakest features until the specified number of features is reachedThe training was done using the R package. The optimisation was done with respect to RFE feature selection method, which was used to select a set of features.

4.6.1 Bootstrap method different datasets

This section uses methods of building ensemble classifiers are demonstrated in the section 4.4. it provides a comparison of the performance of the bootstrap method on seven different UCI repository datasets (See Table 10). Suppose that we have the base classifiers are decision trees of bagging ensemble classifier for the data sets. The trained decision trees classifier generates class predictions in the probabilities form that can be used to evaluate the performance of this approach. Each instance is assigned to the class of the highest probability. The bootstrap samples are 10 times (k=10). The data is split into 70% set for training, 30% set for testing. The criteria of the evaluation of the experiments are average accuracy. The experiments are performed using the R package. We compare the results of the ensemble classifier in term of correctly classified instances (accuracy) on the datasets. Table 15 shows that the bootstrap samples on Thyroid dataset have the highest average accuracy of 99.65 %. Closely following Breast-cancer dataset which reaches an average accuracy of 96.43 %. The accuracy

of 95% correctly classified is provided on the Pima data set. It is noticeable that bootstrap obtains the lowest average accuracy on Indian Liver Patient dataset. It is interesting that the average accuracy result of bootstrap method not only is the best. The poor performance of bootstrap can result from overfitting the training dataset or from instances that are noise [177]. Bootstrap decreases the variance of estimated prediction accuracy at the cost of downward bias (i.e. pessimistic performance estimates are provided by the basic bootstrap). This is corrected by the bootstrap.632, which uses a word for bias correction, and the more modern bootstrap [155].

Table 15: Performance of bootstrap method on five different datasets on the bases of its accuracy

N	Dataset	% Correctly Classified Instances
1	Pima dataset	95
2	Breast-cancer dataset	96.43
3	Indian liver patient dataset	69.23
4	Thyroid dataset	99.65
5	Cardiotocographic dataset	97
6	Hepatitis	94
7	Heart data set	91

The 10-bootstrap routine was performed. at each sample, 70% of the training data were randomly sampled (with replacement) to train decision trees classifier. It is common that real-world data set has missing values. Classification algorithms need complete values in input data set. Therefore, a way has to be applied for handling missing values. Missing values are treated by using multivariate imputation method by Chained Equations (MICE) [35]. We used MICE [184] method to impute the missing values before analysing the data. For example, for used data sets, Hepatitis dataset has missing values in some records. Table 15 reports the accuracy of the ensemble classifier on all seven data sets. The classifier achieving the highest accuracy on thyroid data set. The ensemble classifier performs well on the most data sets used in this experiment. From this finding, we can assume that the ensemble classifier offers performance significantly different on Indian Liver Patient data set. Even if the data has missing values such as Hepatitis data set. There is no significant difference between all the other data sets used. Figure 32 shows the sampling distribution of error rates for decision trees classifier using ten samples. In short, it can be concluded that the ensemble classifier can produce outstanding

performance when considering the measure of accuracy. Whereas, the performance on one dataset was low.

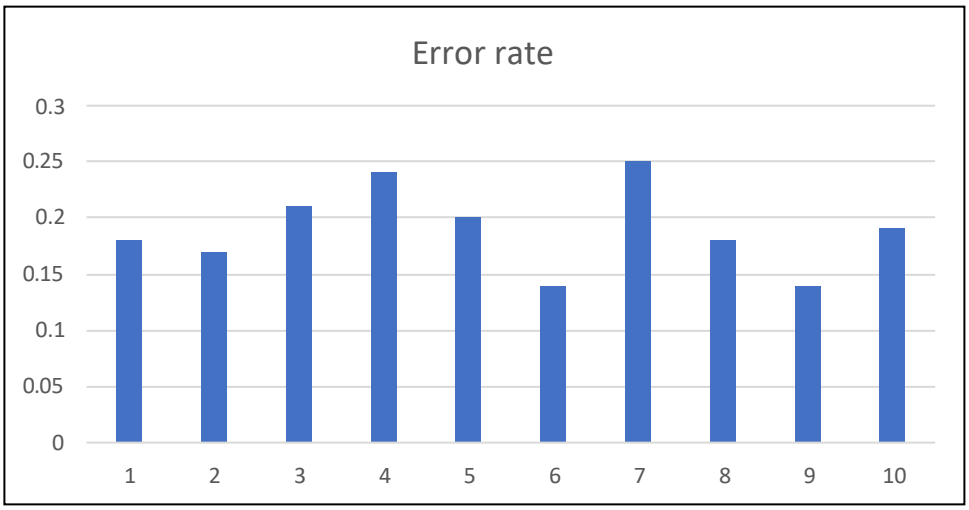


Figure 32: Sampling distribution of Error for heart disease data

According to Brown and Mues, most classification methods can generate results that are quite competitive with each other on given data sets [41]. In the next sections, the obtained decision trees classifier is used to estimate reliability through the ten prediction sets. The STD value is computed using equation 35, and the bias is calculated by equation 34 for each instance in the training set.

4.6.2 The applicability domain of classifiers

Section 4.6.1 provided the comparison of ensemble decision trees classifiers is performed on seven data sets. This section presents the task of assessing the AD of the classifier. For this purpose, consider ten ensemble decision trees classifiers. Consider the case of 2-dimensional input for illustrating the threshold distance of instance. Figure 27 shows the boundaries for variables in 2-dimensional space. It illustrates the effect of correcting neighbourhood distances for their reliability.

For each training set example i, the threshold is computed by equation 27, as previously stated in section 4.5. Since STD is the deviation between a set of outcomes, (1 − STD) represent the precision rate. A high value of precision rate will lead to a significant amount of Wi. Increasing amounts translate into a declining rate of bias in terms of the agreement term, and thus Cov_i will be subject to a small change by a big agreement. Neighbourhood width covered by a specified training point will be fined proportion to its degree of unreliability. For instance, for the value of STD = 60 % and agreement = 30 %, reliability of 12 % will be achieved, resulting in a decrease of 88 % in neighbourhood width attributed to its training example. In contrast, to the high reliability of 96% (STD= 2% and agreement= 97%) will lead to a high value of the neighbourhood width. The neighbourhood width distances for

their reliability is demonstrated in Figure 27. The complete flow of the proposed algorithm is summarised in algorithm 2.

For a given training set D_{train}, the decision of the AD of the model of each test sample J, is:

$$J \in AD\ , I \in D_{test}\, iff\ \exists I \in D_{train} : d_{ij} \leq AD_{all}$$

Taking into consideration that the classification model depends on distinguishing between two distinct reactions and its AD depends on distinguishing between correct and incorrect results. Table 16 shows the summary on datasets depends on the average of the overall thresholds obtained from 100 run times.

Table 16: Summary statistics on datasets based on the average of thresholds

N	Datasets	Accuracy in the AD	The thresholds average
1	Pima Indians Diabetes	0.7432045	0.1670195
2	Breast-cancer dataset	0.9771833	0.3409341
3	Indian Liver Patient data	0.5907089	0.2170842
4	Heart dataset	0.806778	0.4442857
5	Thyroid dataset	0.7028666	0.0796
6	Cardiotocographic dataset	0.9273888	0.413625
7	Hepatitis	0.9086933	0.6416129

The testing set is tested against the training set. We try to determine in this experiment whether testing set examples to fall within the training set neighbourhood. The performance of the classifier at each data point is computed as 1 for cases in the test set correctly predicted and 0 otherwise. Next, searching for training set whose neighbourhood radius includes a test set (which means equal to or greater than the distance to the test data set). The test set falling in at least one training neighbourhood are considered to calculate the accuracy of examples that in the AD of the model, as shown in Figure 27. Figure 33 displays the frequency (group the data into ranges) of the number of thresholds at various values from the minimum amount of threshold to the maximum threshold value. We run the experiment more than 100 times; Figure 28 shows the frequency at the first run time for Pima dataset.

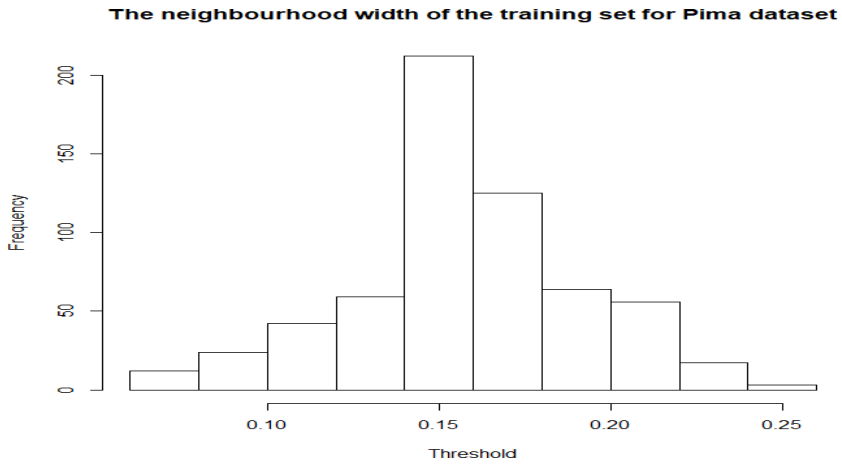


Figure 33: The neighbourhood width of the training set for Pima dataset

To assess whether the test instance falls within the neighbourhood width of training instances. We search for training examples whose neighbourhood width includes test cases. The classifier will test

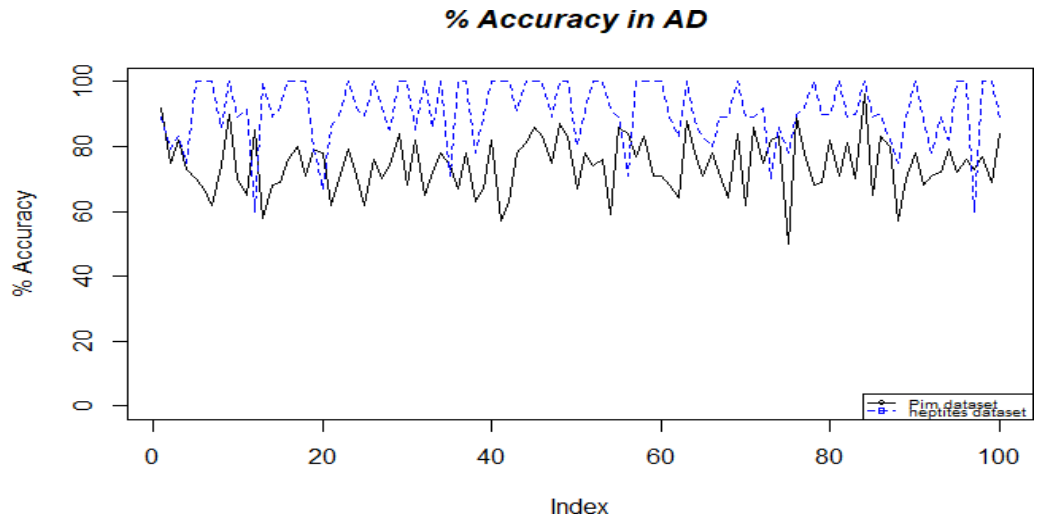


Figure 34: The proposed algorithm applied to Pima dataset and Hepatitis dataset

instances falling into at least one training neighbourhood to calculate the accuracy in AD. The classifier will test instances falling into at least one training neighbourhood to calculate the accuracy in AD. The classifier will give 1 for examples of the test set correctly predicted and 0 otherwise. The proposed algorithm was implemented using all features to assess AD. The classifier performs accuracy between (50%-95%) for Pima dataset, and the range of accuracy for Hepatises dataset is greater values (60%-99%) as shown in Figure 34. Figure 35 illustrates the results of the accuracy in AD which is obtaining by applying the classification method without feature selection methods to heart disease dataset and Brest cancer dataset. The classifier provides an accuracy in range between (57%-89%) forheart disease

dataset, and the range of accuracy for Brest cancer dataset is (94%-98%). The classifier obtained the lowest accuracy rang to Thyroid dataset with (50%-98%), Indian Liver dataset with (50%-73%), and cardiotocographic dataset with (73%-99%).

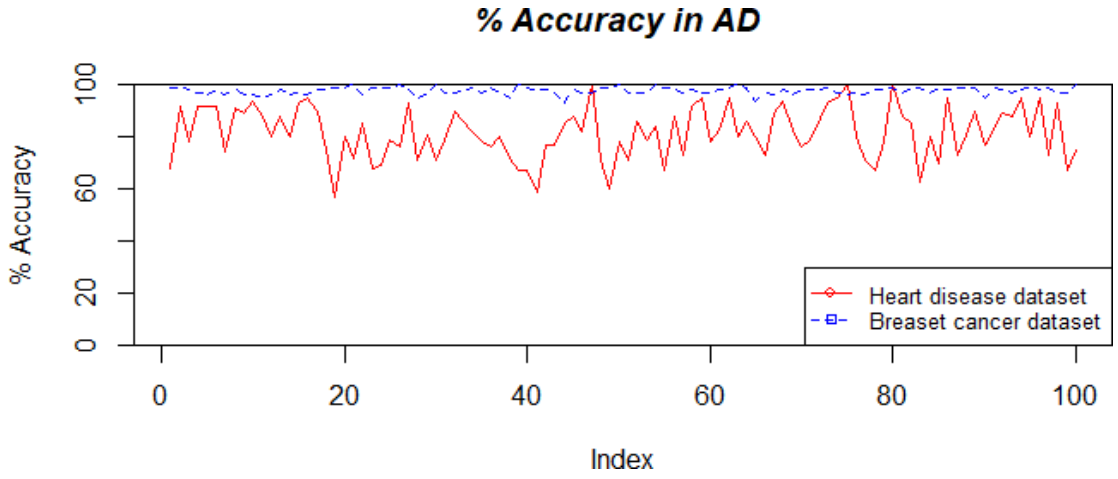


Figure 35: The proposed algorithm applied to heart disease dataset and Brest cancer dataset

Figures 34,35 and 36 show very different curves for different data sets utilised. Interestingly, the model obtained the highest accuracy for Brest cancer dataset and followed by the cardiotocographic dataset. Their accuracies reached 98% correctly classified in AD. However, Table 16 presents the average of the thresholds for Brest cancer dataset and cardiotocographic dataset as 0.3409341 and 0.413625, respectively. The reason for this difference might be computing the threshold locally rather than whole the training set. Table 16 shows the average of the local thresholds of all training set. Aniceto et al established different local thresholds to address the variation of data density across the dataset [115].

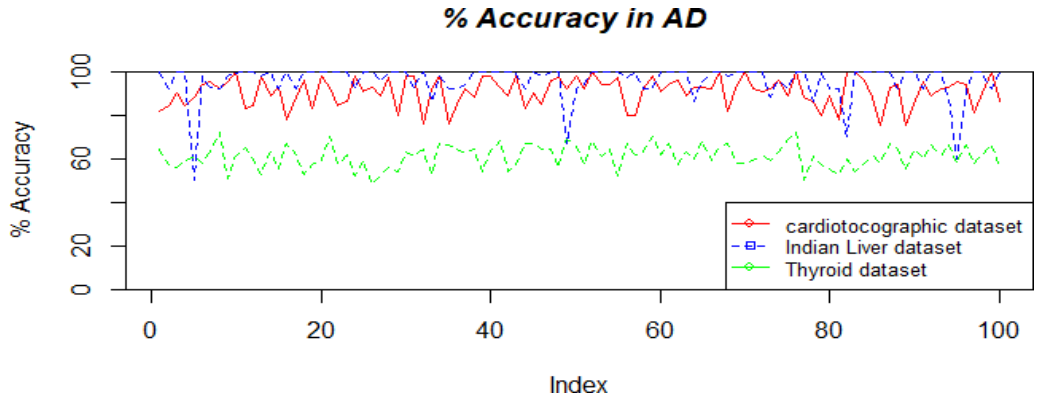


Figure 36: The proposed algorithm applied to Cardiotocographic dataset, Indian Liver dataset and Thyroid dataset

Hepatitis dataset has the average threshold value of 0.6416129. It was the height average value, which is relevant to the distance around the point includes some dataset.

Decision trees classifier was utilised to generate the predictions of the class in probabilities form, which can be used later to assess AD performance. Figure 36 demonstrates the percentage of data test falling within the threshold of instances. It shows the rate of the data that fall in the AD of the classifier for the heart disease dataset, Indian liver dataset, Cardiotocographic dataset, and Brest cancer dataset. Determine whether the test set falls within the neighbourhood of the training set. Figure 37 and Figure 38 show the percentage of data test falling within the threshold of instances for all datasets.

We are looking for a training sample whose neighbourhood radius includes test (new) cases (i.e. is equal to or higher than the range to test examples, as shown Figure 26 in section 4.2). For calculating in-domain accuracy (accuracy in AD) and the data rate (Data in AD), test instances falling in at least one training neighbourhood are considered.

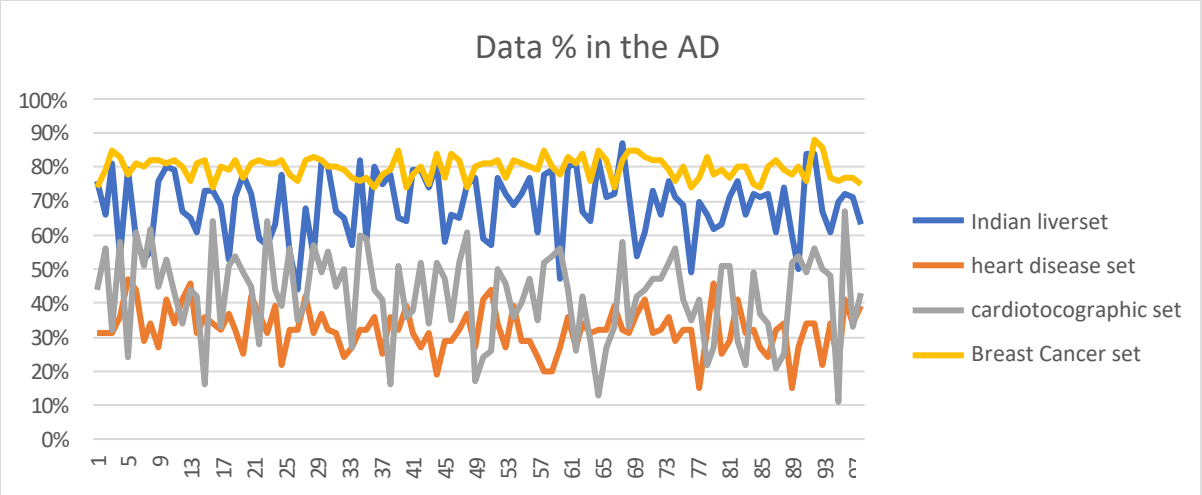


Figure 37: Data % in the AD for heart disease dataset, Indian liver dataset, Cardiotocographic dataset, and Breast cancer dataset

For the model, trying to differentiate between different responses is aimed. Its AD focuses on discriminating between classifications that are correct and incorrect. As earlier proposed [185], it is assumed that the attributes ideally suited for the previous will not necessarily be more suitable for the following time. Another study by F. Sahigara et al showed that the descriptors used to define the boundaries of the model should not match the descriptors used to construct the same model [4], [88], [114], [129]. Additionally, note that the AD method that does not depend on the features used by the QSAR model enables comparable adoption.

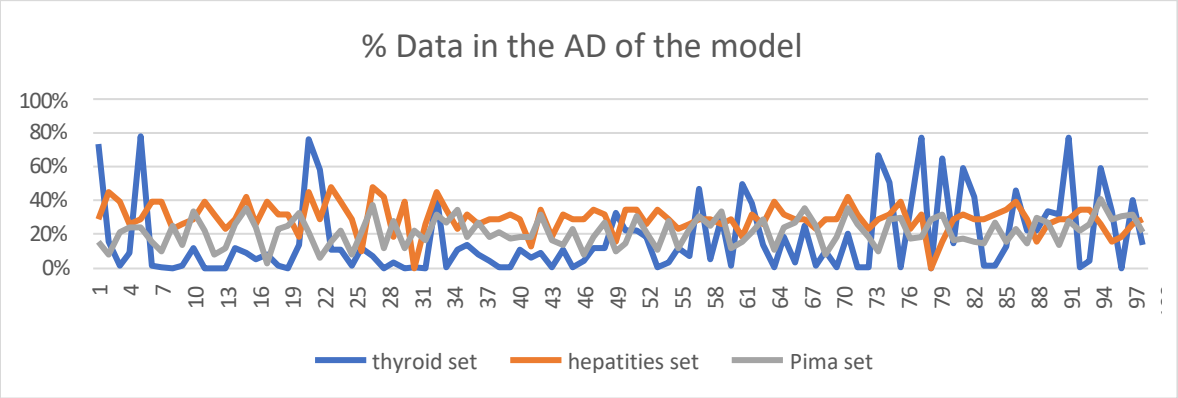


Figure 38: Data % in the AD for Thyroid dataset, hepatitis dataset, and Pima dataset

Transparent techniques (e.g., decision trees) and "black box" techniques (e.g., neural artificial networks). Consequently, the suggested AD technique is combined with the feature selection RFC method. The feature selection method of RFC is used within the building process of the model, as demonstrated in the next section.

4.6.3 Feature selection in applicability domain characterization

Since the individual thresholds obtained in connection with each training case depend on the distance between instances, which in depends in part on the features used, we made a pairing this AD technique with a feature selection method. The proposed algorithm was implemented on seven datasets using the top five features to assess the role of feature selection on AD assessment.

In order to create an optimal feature set used in the algorithm, specifically in calculating the Euclidean distance between the instances in the dataset, distinct feature sets were implemented using recursive feature elimination (RFE) method [185] as demonstrated in the first chapter. Namely the top 5 features as well as the whole set of the features. This resulted in three sets of features being tested in the algorithm. The DT features used to train the model have also been used for comparison, as it is prevalent practice to use the features of the model to define the AD.

Figure 39 illustrates the results of the accuracy in AD of Pima dataset and hepatises dataset. Five features are considered in these experiments after applying RFE method. The classifier provides accuracy between (58%-82%) for Pima dataset, the range of accuracy for hepatises is (59%-98%). Figure 33 demonstrates the percentage of test data that falling within the threshold of instances after applying RFE method for heart disease dataset and Breast cancer dataset. For both datasets, the rate of the data falling in the AD of the classifier are increased.

Table 17: Summary statistics on datasets based on the average of thresholds, after applying feature selection method.

N	Datasets	Accuracy in AD	The thresholds average
1	Pima Indians diabetes	0.7175612	0.1322638
2	Breast cancer dataset	0.9609411	0.2704212
3	Indian liver patient data	0.5907089	0.07263499
4	Heart dataset	0.7840456	0.2568908
5	Thyroid dataset	0.9052205	0.06945
6	Cardiotocographic dataset	0.9275212	0.312125
	Hepatitis	0.8399648	0.3522581

The variance of the data affects the results in some cases. For example, the thyroid data set has man y columns contains only zero and one values. The data has no variation for these columns; thus, the algorithm did not provide a satisfactory performance on this set.

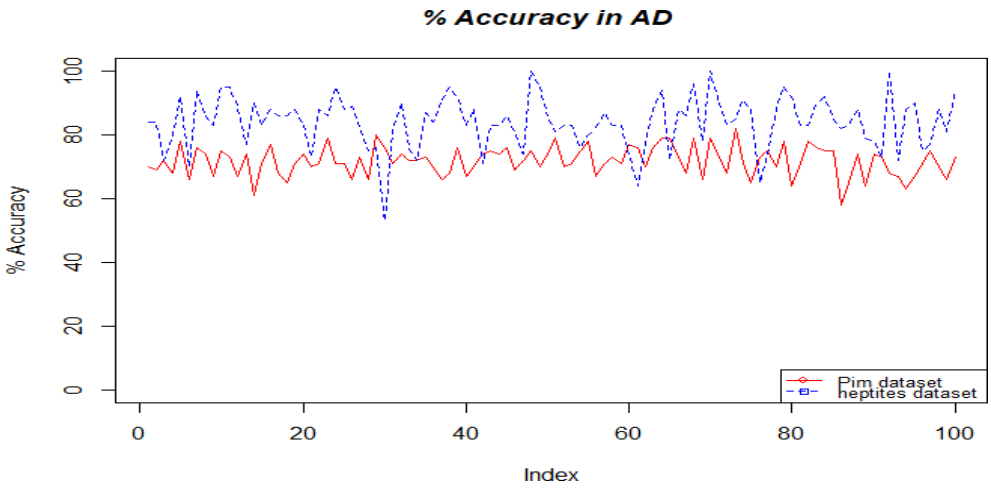


Figure 39: Accuracy Pima dataset, and hepatitis dataset. after applying features selection method

For comparison, table 17 presents the classification average accuracy in the AD, which is obtained by applying the classifier without feature selection methods to the data points that fall in the AD. The average value of the threshold for datasets is included. Table 17 illustrates the classification average accuracy, which is achieved by using the classifier after applying feature selection methods to the data points that fall in the AD. The average value of the threshold for datasets is included.

The results revealed that the accuracy rate in the AD decreased after applying RFE to datasets except for thyroid dataset. The accuracy obtained for thyroid dataset was 0.70 before feature selection, and it became 0.91 after using a feature selection method. The threshold values for datasetsdecreased.

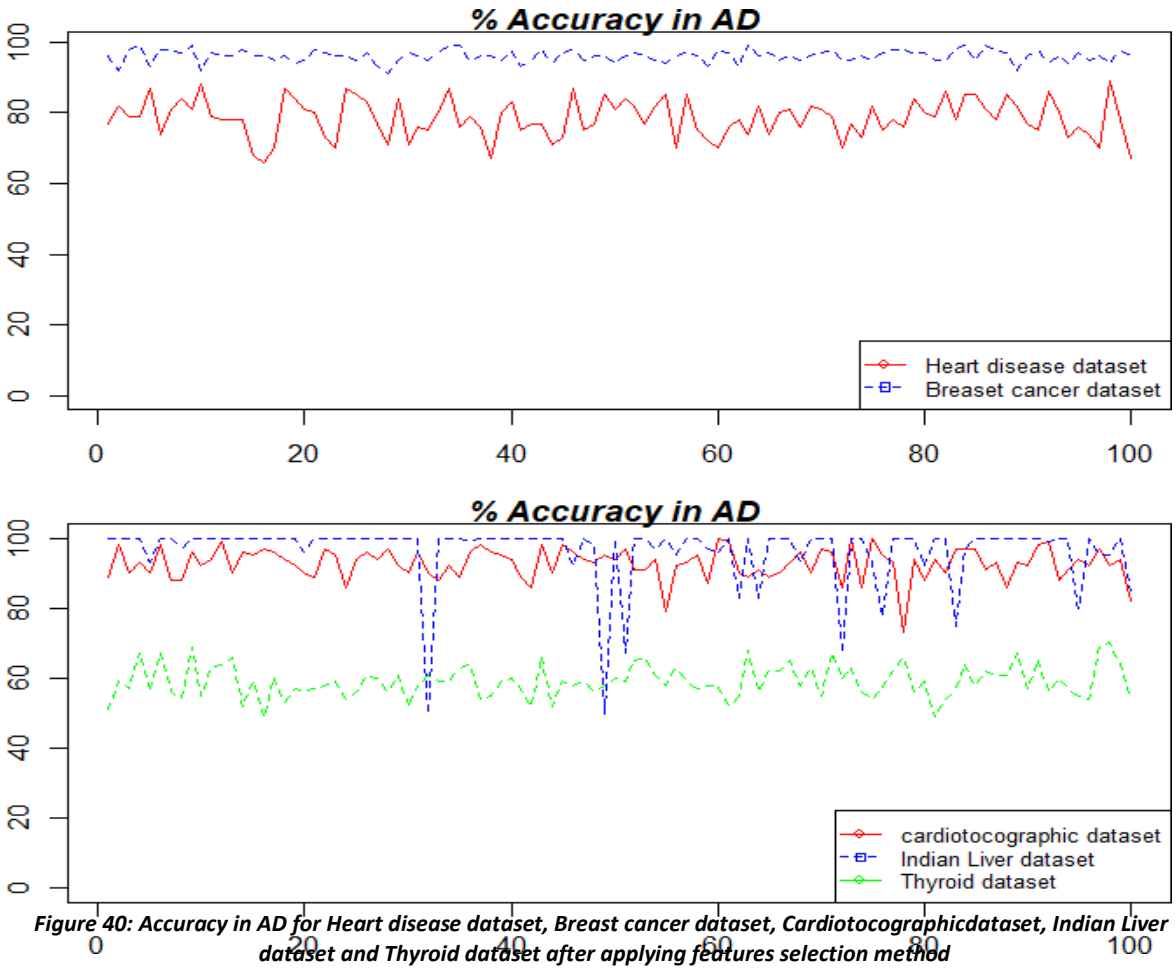


Figure 40: Accuracy in AD for Heart disease dataset, Breast cancer dataset, Cardiotocographicdataset, Indian Liver dataset and Thyroid dataset after applying features selection method

The algorithm was implemented on the datasets using all features and a subset of the features. Figures 34-36, 39, and Figure 40 show different accuracy in the AD curve forvarious features used fordatasets. Accuracy in the AD curve obtained from all features and top five features. Figure 41 shows the percentage of data test falling within the threshold of instances for Pima dataset and Hepatitis dataset after applying feature selection method.

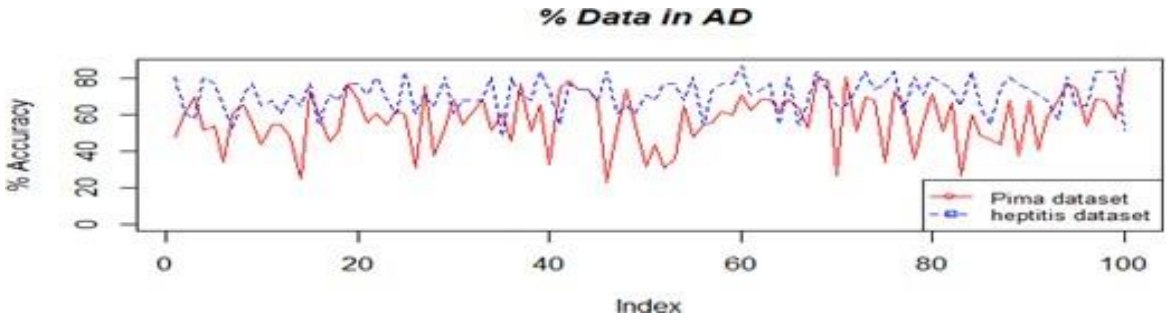


Figure 41: Data percentage in AD for Pima dataset, and hepatitis dataset after applying features selection method

Figure 42 show the percentage of data test falling within the threshold of instances for Cardiotocographic dataset, Thyroid dataset and Indian Liver dataset after applying features selection

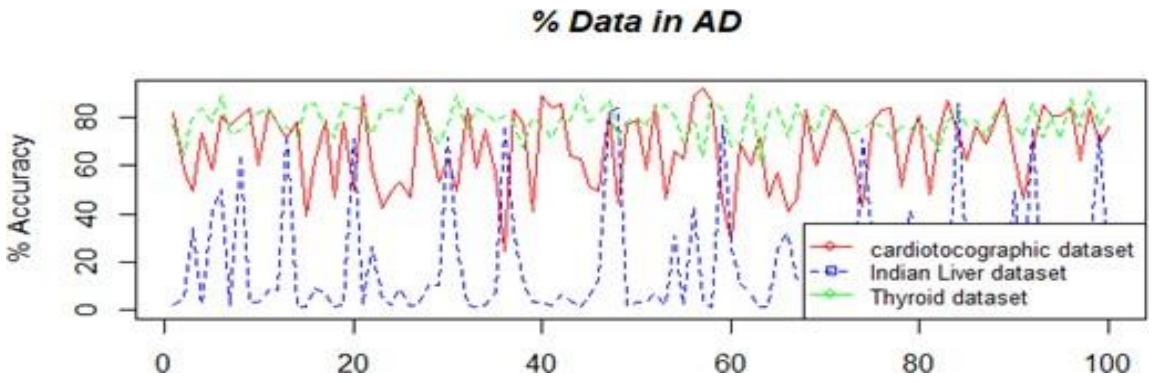


Figure 42: Data percentage in AD for Cardiotocographic dataset, Thyroid dataset and Indian Liver dataset after applying features selection method

method. The findings stated that some classification techniques work well without the application of feature selection methods and improved performance obtained by the classification model when all characteristics are present in some datasets. Due to no impact of feature selection methods on classifier performance of this proposed approach, for some datasets the classification model did not produce different outcomes such as accuracy in AD of Breast cancer dataset (See Figure 44). However, some datasets have higher results after implementing feature selection methods such as Pima dataset.

4.7 Summary

The classification models for machine learning depends on the theoretical assumption of the relationship between independent attributes and the dependent attribute. It is crucial to identify the regions that the classification model is safely used. In this section of this work, we covered the AD

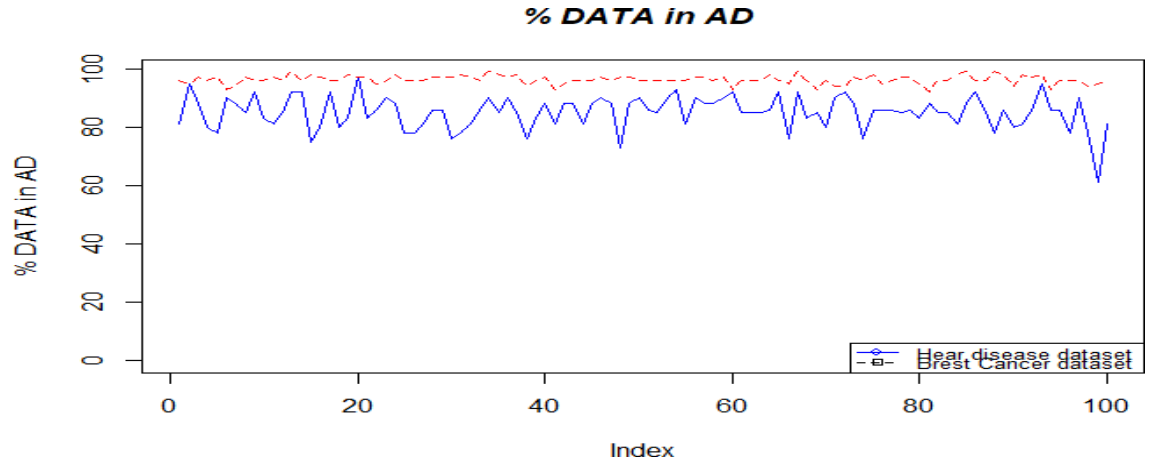


Figure 43: Data percentage in AD for Heart diseases dataset and Breast cancer dataset after applying features selection method

evaluation of classifier, comparison of ensemble classifier based on decision trees model on ten healthcare datasets, the effect of FS methods on the AD assessment of classifier. First, Bootstrap method on five different datasets, second assessing the AD based on the precision and the agreement of the classifiers, third, investigation of the effect of feature selection method on The AD estimation. We looked at an ensemble classifier of decision trees classifiers in this section and studied their efficiency over seven real-life data sets of healthcare. Six datasets used to be analysed in the context of correctly classified, i.e. Pima Indians Diabetes, Breast-cancer dataset, Indian Liver Patient data, Heart dataset, Thyroid dataset, Cardiotocographic dataset, and Hepatitis dataset, were also selected to provide a broader evaluation of available method on given datasets. Based on the measure of accuracy, the classification capacity of the ensemble classifier method was evaluated.

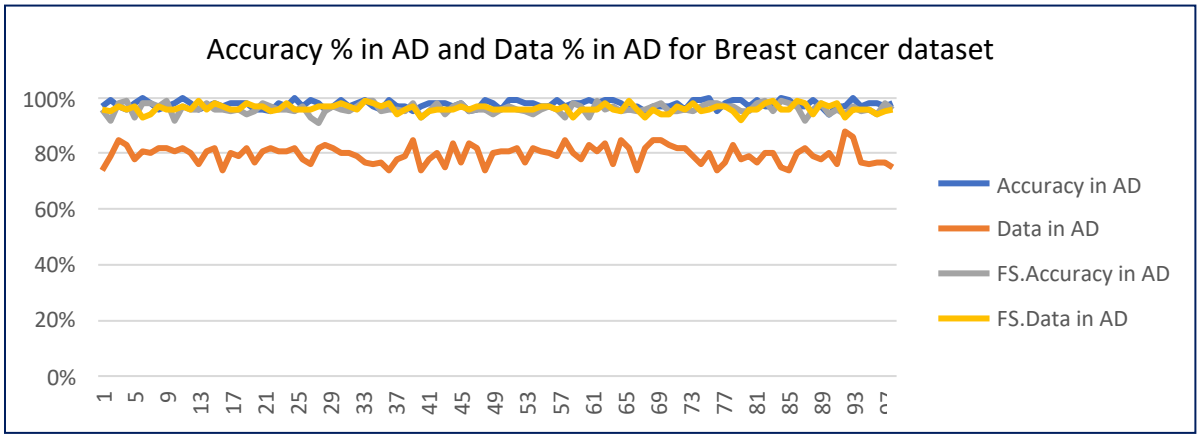


Figure 44:: Accuracy % in AD and Data % in AD for Breast cancer dataset

Finally, the features of the data that are being used to train machine learning models have an impact on performance. Therefore, the feature selection process in machine learning can influence the performance of the ML model hugely. Figure 44 shows the performance of the classifier on Breast cancer dataset as an example of how the accuracy and the data rate can be affected. Moreover, it is to show how feature selection method can make a difference result. It presents a comparison between the performance with entire features and the results after applying FS method. The findings stated that some classification techniques work well without the application of feature selection methods and improved performance obtained by the classification model when all characteristics are present in some datasets. Due to no impact of feature selection methods on classifier performance of this proposed approach, for some datasets the classification model did not produce different outcomes such as accuracy in AD of Breast cancer dataset (See Figure 44). However, some datasets have higher results after implementing feature selection methods such as Pima dataset.

5 Robustness of classification models based on applicability domain approach

5.1 Introduction

Machine learning techniques integrate artificial intelligence (AI) algorithms for extracting patterns learned from the data. This process is known as learning (or training) stage [28]. Most used machine learning techniques include ANNs [84], SVMs [32], and decision trees (DTs) [28]. However, most of these algorithms just focus on how to improve the classification performance, while the robustness is not taken into consideration. Currently, data is collected at an extraordinary rate and from different and diverse sources in many domains and disciplines. The ML techniques are widely used in various domains for data-driven decision support and components in more complex knowledge-based expert systems. Particularly, artificial intelligence and machine learning are being used in several applications such as image recognition [90], natural language processing [186] and control systems [187]. Mainly, ML has branched into supervised learning and unsupervised learning. In this work, we consider the approach of supervised machine learning classification. Classification technique can be defined as the task to predict a class label for a new given test dataset [84].

A large amount of data about individuals and organisations become available in various fields. Such data can be used to understanding and planning the processes in sectors. This data allows public and private sectors to record a considerable number of relevant datasets. This volume of data being gathered is not possible for analysing by human beings [116]. The estimation of machine learning algorithm robustness is a real-world problem in many other areas.

The estimation of machine learning algorithm robustness and the model applicability are an essential task and sometimes result in inefficient performance. The reason may include lack of knowledge about the model capability [114]. The applicability domain of the model has become to be a significant

challenge within the machine learning field because the AD of a model is to assess the ability of the model to determine whether new data satisfies the assumptions of the model [4][113].
The level of generalisation of a given predictive model can be determined by defining its applicability domain AD. In this way, if the AD is too restricted, it means the model expectations can be very limit. According to [115], the Quantitative Structure-Activity Relationship (QSAR) model should have a definition of applicability domain (AD) and appropriate measures for goodness-of-fit, and robustness. The robustness of the predictions from such models has been demonstrated in [141] as a growing concern. Even though some models have high accuracy as carried out in [113], it is not useful to ignore determining where the model can provide reliable results or not. Robustness is a measure utilised in differing situations for machine learning models, for example, the capacity of the classifier to make right predictions on the noisy dataset or a dataset with missing values. AUC measures have been accounted for improving the quality of robustness of classifiers in terms of measuring high sensitivity or correct positive rate [17][25].

The goal of this work is to evaluate the robustness of machine learning classification techniques in giving proper outcome with a specific dataset by identifying its applicability domain. Mainly, the experiments focus on the evaluation of the performance of random forests classifier by using many generated datasets as input data over various operations. First, random forests classifier is compared with the five well-known classification algorithms. Second, different sets of input datasets in the classification system are studied. Finally, the robustness of the classifier is checked based on generated data sets. For the modelling task, a response variable can be used to assess the performance of the model. The main question is, can AD be useful in identifying the robustness or the applicability of the ML model? In the experiments of this work, healthcare datasets have been considered as an example for analysis. All studied datasets are publicly available on [145]. Random forests classifier has been built and evaluated using the R package, and the performance metric is the accuracy. Machine learning models in healthcare data assess the health status of the patients to aid in clinical decision-making, including results, inpatient admission or diagnosis based on electronic healthcare records. In the healthcare domain, all the information will be stored in electronic health records [188].

In some cases, life can be saved if the model can be checked. What if in some situations, the model does not provide proper results. Therefore, the patient discharged from the hospital. People heath can affect because the model is not robust enough, resulting from the lack of a way to define the applicability domain of the model.

5.2 Procedure of the algorithm

A sequence of essential tasks needs to be performed for defining the applicability domain of the classification model. To efficiently perform these tasks, a set of steps are presented as follows:

1. Prepare the dataset by removing the outliers and Normalizing the data set.

Outliers are extreme values in the data that can affectthe classifier performance [28] because a single substantial amount can skew the sample average. A small fraction of absolute values is discarded from the data by using outlier detection techniques in the preparation stage of building the classifier [189]. All data features will take values in the range [0,1]., each value is scaled by:

$$x_i = \frac{x_i - \min(x)}{\max(x) - \min(x)} \quad (2)$$

where, x is a set of observations, $min(x)$ is the minimum value for variable x and $max(x)$ is the maximum value for variable $x$and x_i is normalized data.

2. Split to training set D_{train} and test set D_{test} .

The dataset is split into training sample D_{train} (70%) and testing samples D_{test} (30%). Next, the training set is used to train a random forests classifier and evaluated based on data test D_{test} .

3. Compute the distance matrix $\delta(x_i, x_j)$.

The distances of each sample of the training dataset D_{train} from the rest of the examples of the training set are calculated by using the R function rdist [160]. The distance of features vector $(x_1, x_1, \dots . x_n) \in R^n$ can be calculated by measures such as Euclidean metric, Manhattan, or Minkowski. The most used method is the Euclidean method, which defined as

$$d = \delta(D_{train\,i}, D_{train\,j}) = \|D_{train\,i} - D_{train\,j}\|_2 \quad (3)$$

Where, d is the distance between ith and jth points in D_{train}. The number of values within d is equal to $n(n-1)$ without zero values.

Make an increasing order of the distance matrix d ($d_{i1} \le d_{i2} \le \dots \le d_{in-1}$). d is used to obtain a reference value (upper limit) set at $Q_3 + 1.5 \times IQR$. It is Tukey Fences method [190]. It is explained

in section 4.2. The threshold t for each training sample is calculated as the average distance to all its training neighbours with distance values equal to or more than the upper limit.

If the distance value of the i-th sample from its given j-th data set neighbour (where $1 \leq i \leq$ n–1) is less than or equal to the MAXdist and; more than or the same t, then that distance value is retained, $t \leq d \leq MAXdis$. Otherwise is neglected (See Algorithm 3 and line 9 in Algorithm 1). The points replaced in the range between a reference value t and the maximum value of the distance are only chosen. t is defined in table 1 as a close value to MAXdist. Sub_t contains all training samples with distance values closer or equal to MAXdist.

4. The distance between a given data set Sub_t and the training data sets d_{train} is determined and compared simultaneously.

When the condition remains valid for at least one training data set, the test sample will be considered inside the space of that model Sub_{wthin}. Otherwise, the expectation for that test data set will be outside the space $AD_{outside}$.

$$Sub_t \in Sub_{inside} \exists\ i \in D_{train}\ ; \max \{\delta(x, y): x \in Sub_t, y \in D_{train}\}$$

The distances between Sub_t and D_{train} is $\delta(x, y)$.

5. Each i-th data sample is allocated in the required range was selected. Sub_t is defined as: $t \leq Sub_t \leq MAXdist$, which $Sub_t \leq Sub_{test}$.
6. Add a random number r between 0 and 1 ($0 \leq r \leq 1$) to each element in Sub_t to obtain a new unlabeled data set which probably contains points allocated in the border of the train data set. After that operation, we get the new sub-dataset. Prediction of labels of the new data set can be performed by vote with all built classifier's predictions [169]. Theoretically, r value can affect the result of the classifier. Figure 5 shows the number and rate of points that are inside and outside the domain of the model with r value.
7. Make predictions on the point from test set that falls in the range of Sub_t and check the points that have correct and incorrect predictions. The model will be robust at some points and not at others. This may mean the point has good accuracy falls inside the domain of this model. On the other hand, the point has poor accuracy lies outside the domain of the model.

We have described the general workflow of this approach in three stages, as shown in Figure 27 in page 67. Summary of notation of algorithm 3 is shown in Table 12 in page 65.

5.3 Experiments and Results

In this section, first we present the experiments description and the results of the proposed algorithm.

5.3.1 Experiments description

On these datasets, algorithm 3 was applied to demonstrate efficiency. We run the experiments for the parameters and the datasets. The robustness of the model is evaluated based on the prediction of the model or the ability of prediction. Moreover, the capacity of an assessed model should be estimated before using the model practically [109][191]. The test set will involve new extra data created later. These data are synthetic, and they are generated after the model has been fitted. If this the new test data set fall inside the model space effectively, they should be represented to confirm the validity of accepted models, based on the distribution in the model area. By the guide of synthetic test data points, it is possible to assess whether a model evaluated from the training data set is also a proper model for future elements. To illustrate the idea of this work, let us consider the robustness of the model that shows a stable loss of accuracy with increasing distance to the central points where the predictive confidence is maximum.

We illustrate the proposed algorithm on the datasets from table 9, using the RF classifier with use $r = (0.10, \ldots, 0.20)$.

5.3.2 Results

Evaluating the performance of the RF classifier on the data sets is given in Figure 45. A brief description of the datasets is presented in Table 10. Some datasets are pre-processed before performing the experiments such as removing missing values.

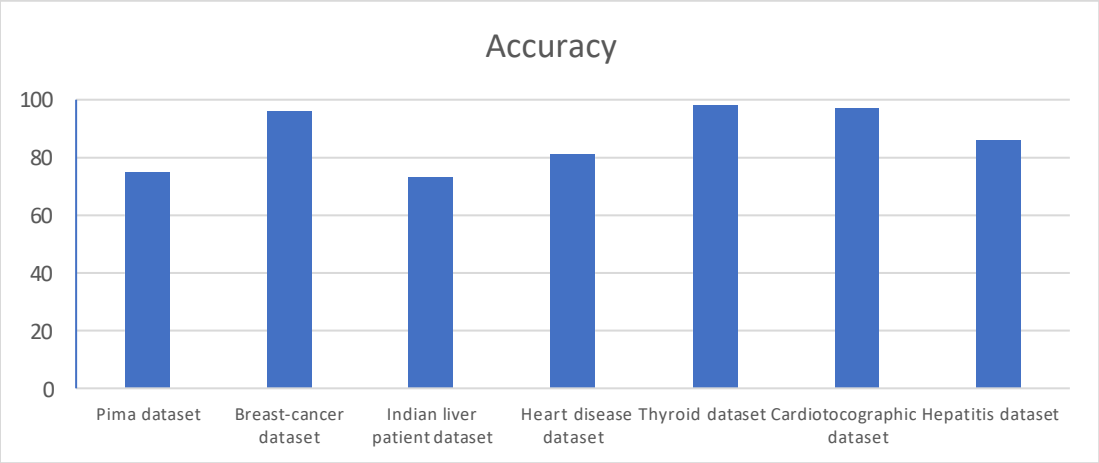


Figure 45: The performance of RF on datasets

10-fold cross-validation method is used for the classification performance. The RF algorithm provides the here highest accuracies for Thyroid dataset (reached 98%.), Cardiotocographic dataset (reached 97%.) and Breast-cancer dataset (reached 96%.). However, Random Forests provides the lower accuracy for Indian liver data set (reached 73%.).

By obtained results, the robustness of the model can be described as the AD of a model in the space. Table 18 displays the number and rate of points that are inside and outside the domain of the model with Pima dataset. It contains the value of r (0.10 - 0.20), the subset that is inside the domain (Inside#), the subset that is outside the domain (Outside#), rate of data points that are inside the domain (Inside%), and rate of data points that are outside the domain (Outside%). By adding r=0.01, the number and the rate of the points that are located inside decreased each time. However, the quantity and the rate of the points that are outside increased. As previously explained. The points from the test data set that full in the same area with the new point are considered each time we added r to the border points from the training set. It is described in line 13 from algorithm 3. We can evaluate the model on only the considered points from the test set (labelled points).

Table 18: Number and rate of points that are inside and outside the domain of the model with Pima dataset

r	Inside#	Outside#	Inside %	Outside%
0.10	116	32	78	22
0.11	111	37	75	25
0.12	105	43	71	29
0.13	101	47	68	32
0.14	93	55	63	37
0.15	86	62	58	42
0.16	83	65	56	44
0.17	78	70	53	47
0.18	75	73	51	49
0.19	69	79	47	53
0.20	66	82	45	55

After calculating the distance matrix of the training set and determining the maximum distance between all the points, we chose the point they have maximum values Sub_t. The total number of the testing set is 148 instances. We add the first value of r (r=0,10) to Sub_t. The number of points from test set that fall in the range of chosen points Sub_t is 116 data points (78%), and the number of 32

instances (22%) is out of the range. At the value of r equal to 0.11, the number of points from test set that fall in the range of chosen points Sub_t decreased to 111 data points (75%), and the number of 37 instances (25%) is out of the range.

We compared all the results according to the number of the correctly classified and the classifiers accuracies. The classifier gives the highest number of correctly classified (0.78) at the first value of r. The low number of examples correctly classified at the last value of r (r=0.20) (see Table 19).

Table 19: Comparison of r value with RF classifier on Pima dataset according to the accuracy, correct classified classes, False positive, and False negative

r value	Accuracy	Correct class	False positive	False negative	Number of Instances
0.1	0.78	116	23	9	148
0.11	0.78	111	23	8	142
0.12	0.77	105	23	8	136
0.13	0.77	101	22	8	131
0.14	0.76	93	21	8	122
0.15	0.75	86	20	8	114
0.16	0.76	83	18	8	109
0.17	0.76	78	16	8	102
0.18	0.76	75	16	8	99
0.19	0.75	69	15	8	92
0.2	0.75	66	14	8	88

The proposed approach was implemented in five datasets (from Table 10) using different values of r to assess the impact of different values of the threshold r. Figure 39 shows a very different curve for different r used. It shows the rate of points that are inside the domain of the model with r for the five datasets. The number of points on the vertical axis and the value of r is on the horizontal axis. r has the sequence values between 0.1 and 0.2 by increment 0.1. Each time we add r to the training set, we test the points from the test set that fall in the same area (inside AD) with the training set. Applying the classifier on only these points yields the following results:

The accuracy is between 90% and 80% for all data points. In this study, all the results were compared based on a number of the correctly classified and the classifiers accuracies on all datasets (See Table 21). The r values are 0.10 and 0.20.

Table 20: The accuracy of RF on the points that are inside the domain of the model with Pima dataset

r	Inside#	Accuracy%
0.10	116	85
0.11	111	86
0.12	105	90
0.13	101	87
0.14	93	89
0.15	86	81
0.16	83	88
0.17	78	83
0.18	75	80
0.19	69	81
0.20	66	80

RF classifier gives the highest number of correctly classified on Thyroid dataset for both values of r. The accuracy is low of correctly classified on Indian liver patient data for both r values. For the rest of the data sets, the accuracy at r=0.10 is higher than the accuracy at r=0.20.

Table 21: Comparison of r value with RF classifier on all dataset according to the accuracy, correct classified classes, False positive, and False negative

Datasets	r value	Accuracy	Correct class	False positive	False negative	Number of Instances
Pima indians diabetes	0.10	0.78	116	23	9	148
	0.20	0.75	66	14	8	88
Breast-cancer dataset	0.10	0.84	26	3	2	31
	0.20	0.81	17	2	1	20
Indian liver patient data	0.10	0.62	20	9	3	32
	0.20	0.67	10	5	0	15
Heart disease dataset	0.10	0.75	18	6	0	24
	0.20	0.63	5	3	0	8
Thyroid dataset	0.10	0.99	123	0	1	124
	0.20	0.99	74	0	1	75

The most important question for us was how the different values of threshold r could affect the outcomes. With these initial values of r, the algorithm takes various iterations to converge.

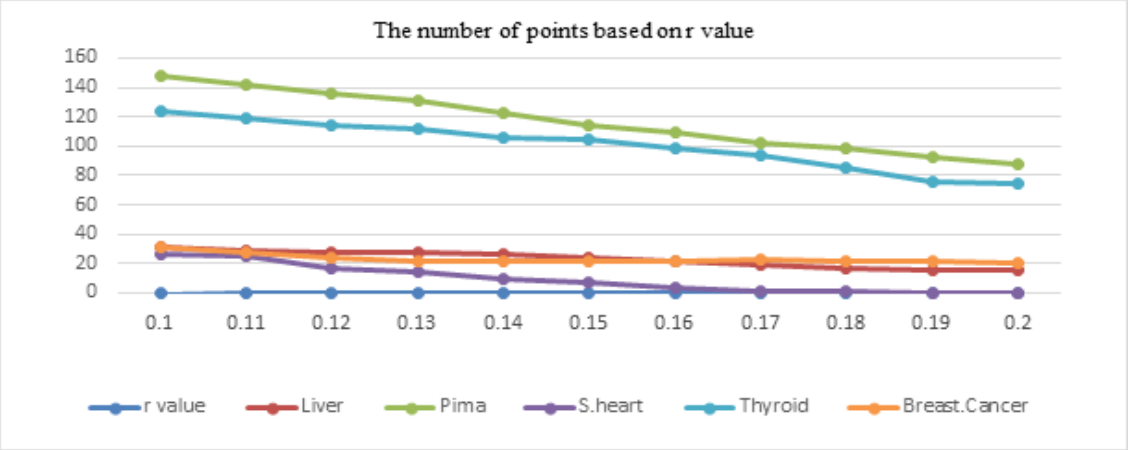


Figure 46: Comparison of different five data sets used. The rate of points that are inside the domainof the model with r

Figure 46 shows the amount of r vs the number of instances, and it displays a very similar trend between the data sets. There are a few cases drop at each value of r we added, which corresponds to a specific Euclidean distance from the training instance. So, it is probable that the test space corresponding to instances that fall around this distance is a way to define the robustness of the model.

Consequently, more important than having a perfect rate of accuracy concerning the overall performance of the model, it is a priority that the algorithm can correctly describe how new data will behave robustly across the dataset space.

5.3.3 Discussion

We study a concept of the behaviour of the ML model, called the robustness of the ML model, defined by several authors. We consider binary and multi-classification. L.Hellerstein has reported that the on-line algorithms in the context of deterministic binary classification [192]. They defined the robustness of the classification model using the p-norm algorithms to achieve robustness when applied to learn noisy data. Kanamori et al. [40] proposed a unified formulation of robust learning methods for classification and regression problems. They used the hinge loss with outlier indicators for detecting outliers in the observed data to define the robustness property. In another study, Hines et al. proposed a unified formulation of robust learning methods for classification and regression problems. They used the hinge loss with outlier indicators for detecting outliers in the observed data to define the

robustness property. In another study, Hines et al. [112] discussed the robustness of ML in supporting objective quality assessment is studied, especially when the feature set adopted for prediction is suboptimal and assess their adaption with noise. Ghosh et al. [37] investigated the robust of a learning algorithm to label noise. They present some theoretical analysis for popular decision tree algorithms to show the robust to symmetric label noise under large sample size. They also offer some sample complexity results that provide some bounds on the sample size for the robustness to hold with a high probability. Hu and Tan [192] performed an approach to analyse the robustness of four well-known machine learning-based malware detection approaches, i.e. the DLL and API feature, the string feature, PE-Miner and the byte level N-Gram feature. They proposed two pretence approaches under which malware can pretend to be benign and bypass the detection algorithms. Pelletier et al. evaluated the robustness of random forests classifier based on improving the classification accuracy. They mapped land cover with high-resolution satellite image time series over large areas [109]. Shami and Verhelst [31] provided an approach to construct the classifier based on the merged databases of emotional speech datasets, which recorded under different conditions. Thus, the classifier is more robust than a classifier learned on a single corpus. Here we described another way to use the applicability domain approach to evaluate the robustness behaviour of the classifier. As we mentioned above in section 3.3.2, the predictive ability of the model reflected by using the accuracy and classification error. Since the data points that have poor accuracy are considered as unreliable predicted (outside the domain of the model). The results of the model are represented in two cases:
The first case: the rate of samples that have good accuracy.
The second case: the rate of examples that have reduced accuracy.
Consequently, the robustness of the model is evaluated based on the ability to provide a good performance of test data point that falls around the training dataset space border.
The results obtained from this study provided an understanding of how the proposed approach can help to define the model's robustness and the applicability domain, for providing reliable outputs. These approaches open opportunities for classification data and model management. The proposed algorithms are implemented, tested and validated through a set of experiments, a classification accuracy of instances that fall in the domain of the model. For the robustness of the classification model based on the applicability domain approach, the minimum accuracy is 0.62% for Indian Liver Patient data at r=0.10, and the maximum accuracy is 0.99% for Thyroid dataset at r=0.10.
In machine learning, the weight of the learning process can be started with a small random value such as in neural network technique. In the neural network, a small random value of parameters to be chosen in and then these values are changed based on the feedback or an optimization process.

The value of r is a weight that allows creating a halo of the AD. We follow a similar approach in ML such as the weight in the neural network [50]. In future work, we are going to look at how this number to be generated to optimize the cost function to be done.
Here we choose a small number to demonstrate the concept of AD. However, this number should be optimized by following some criteria related to the optimization of a cost function. The r value in this study is to have the best accuracy. Choose this number to maintain accuracy. The choice of r is related to a cost function that optimizes the loss of accuracy. For simplicity, we choose a small number to see how the concept development. However, more researches are necessary to link the choice of r to the optimization of a function related to maintaining the accuracy of the model.

5.4 Summary

The k-nearest neighbour algorithm is susceptible to the local data structure, and the existence of noisy or irrelevant features affects its performance. The best selecting of k depends upon the data. Larger values of k tend to decrease the impact of noise on the classification. However, it will make less distinction between classes. Cross-validation can help to select a good value of k. It is useful to choose k as an odd number in binary classification problems as this prevents tied votes.

The overall implementation of this approach was performed in three different stages that efficiently used the features of the AD concept to define a model's robustness. The significant features of this proposed contribution include firstly, measuring the distances to identify the close points. Secondly, using synthetic data to test the robustness of the model. Thirdly, defining the threshold for each test data, and Finally, optimising the threshold parameter r. The distance matrix is considered and considering the model's response domain to reflect upon the reliability of results derived in its descriptor space. This proposed method identifies an appropriate random value where the model accuracy changes to define the robustness in the model. This approach was applied successfully in some benchmark datasets, but it has limitation for small datasets. The method can be applied in a wide range of domains such as Education or Economy. Using different ways to generate synthetic data may affect the results. Uncertainty can derive from many sources, including incomplete observability and incomplete modelling. The most important question for us was how the different values of threshold r could affect the outcomes. Further work that could be conducted, as a result of this study, additional research opportunities in the field of investigating machine learning algorithms to deal with uncertainty using probability theory. Further research would be required that effect on machine learning model robustness and beneficial for measuring and characterising applicability domains and the confidence in the results.

6 Defining the optimum classifier

6.1 Introduction

This section presents the process of machine learning model selection for classification models. An algorithm is performed based on Pareto optimality to mines a set of classification models to select a reliable model that offers reliable results for a test dataset. After getting the ensemble classifier from the approach of assessing the AD for classifiers in Chapter 4, the available collections of models are used for finding a better model among them. A multi-optimization problem approach can be defined as the problem of finding the Pareto set. In this section, we used the application of Pareto optimisation (PO) in solving the model ability (applicability domain). The main goal of offering the multi-objective optimisation problem is to maximise model accuracy and threshold. To select a model for a given test set we convert the set of model's performances into a set of pairs (ACC, Ta), where ACC represents the accuracy of the classifier and Ta is the threshold.

6.2 Multi-objective optimization model

In some cases, the critical need for efficient classifiers creates a costly and complicated building of machine learning models. In this respect, optimisation modelling can be a powerful tool to solve the model's reliability problems, and studies have focused on improving classifier performance [164][193][25]. However, due to a large amount of uncertainty and the complexity of models in post-build operations, previous optimisation models have reflected the single objective [194][195].

In a study [193], a method for ML algorithm was proposed. They reviewed 100 UCI repository classification problems in research and assessed eight various data mining classifiers based on a combination performance measurement of average accuracy and computation time.

Assume that a collection of classifiers produced by the dataset. It is essential to mention that these classifiers are required as inputs into the MO model. The two objectives used are shown in Table 23. Note that these have been converted into maximisation objectives for optimisation.

Table 22: Performance objectives

Objective 1	Classified correctly	ACC.IN.AD	ACC
Objective 3	Thresholds average	TRS.AD	Ta

Table 23 shows some random solution of the optimisation process; next, all objectives are evaluated and sorted to find the Pareto set of optimal solutions from the feasible solutions. Table 24 represents the values of accuracy (ACC) and Thresholds average (Ta) of each model, and the goal is to find the optimal solution by comparing the solutions based on the objectives. The optimal solution will have a maximum value of ACC and Ta (See Table 25). the optimal solutions based on the Pareto concept are non-dominated by other solutions (not comparable) that lie on optimal Pareto front. Multi-criteria of 10 models for heart disease dataset is illustrated graphically in Figure 48.

Table 23: some random solution of the optimization process

ID	# ACC in AD	# Data in AD	# Thresholds average
1	0.888889	0.202247	0.319395
2	0.916667	0.134831	0.345668
3	0.8125	0.179775	0.328117
4	0.933333	0.168539	0.325106
5	0.869565	0.258427	0.336872
6	0.785714	0.157303	0.320966
7	0.9	0.11236	0.319825
8	0.782609	0.258427	0.331974
9	0.714286	0.157303	0.320125
10	0.875	0.179775	0.316321

The objectives of the MO optimisation model are to maximise accuracy in AD and threshold. To solve the model, we produce an approximate set of Pareto optimal solutions where each solution represents a classifier, which may have different values of accuracy in AD and threshold.

Each classifier is characterised by the selection of (1) data rate more than zero, (2) accuracy higher than 50%, and (3) threshold equal or more than the average of the threshold of all classifiers. Objectives seem to be not conflicting, because usually, for these types of problems, a classifier with higher accuracy involves higher data rate that falls in the AD of the classifier and vice versa.

$$ACC = \{\, ACC \in F \mid ACC \geq 50 \,\} \tag{37}$$

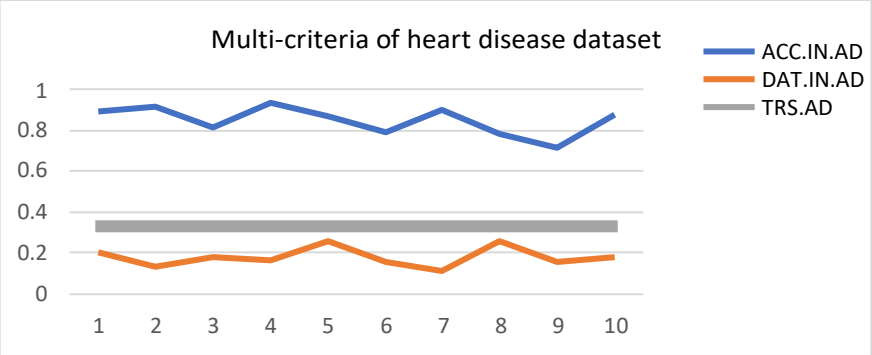


Figure 47: Multi criteria of 10 models for heart disease dataset

As the solutions evolved, most of them converged toward the value more than the accuracy of 60% (correctly classified). Interestingly, some solutions with more than 90 of accuracy appeared in the Pareto set in the earlier generations. However, these solutions led too low threshold value. Consequently, in the latter Pareto set, new solutions with reasonable accuracy have replaced the old solutions with high accuracy. However, they incur lower threshold and deliver the same or even more accuracy.

According to the results, table 24 presents the characteristics of some of the Pareto solutions. For instance, Solution number 38 outperforms others concerning both criteria. It has an accuracy of 0.9444444, 0.3499654 of the thresholds, and. It is represented as a dark black point in the top of the plot in Figure 49. Five classifiers (numbers 7,37,50,58 and 69) were the worst performance based on both criteria. They are in the bottom plot in Figure 48 (dark black points). The rest of the points in the plot in Figure 49 represent the generations of Pareto point between the best solution and the worse solution.

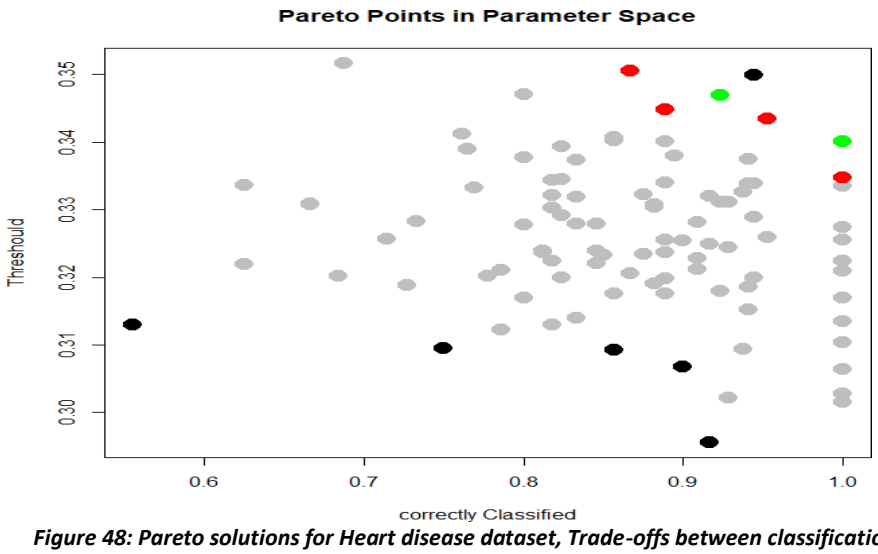


Figure 48: Pareto solutions for Heart disease dataset, Trade-offs between classification objectives

The total number of feasible solutions was calculated. A few the original solutions (feasible and not-feasible solutions) that do not satisfy the constraints were removed. This analysis required testing all solutions to discover the solution space.

Table 24: Samples of Pareto solutions

N	Classifier number	ACC	T
1	**38**	**0.9444444**	**0.3499654**
2	22	0.9230769	0.3469547
3	7	0.9000000	0.3066990
4	37	0.5555556	0.3129616
5	50	0.8571429	0.3091899
6	58	0.9166667	0.2956086
7	69	0.7500000	0.3094657

Table 25 presents the generations that discovered the Pareto solutions and summarises the percentage of Pareto solutions found at each generation. Clearly, in later generations, less of the Pareto solutions are obtained.

6.3 Discussion

The experiments were done to demonstrate the advantages of using pareto points for model selection. For each generation, better solutions are created and introduced into the Pareto set. Therefore, most of the solutions in the final Pareto set have been found in recent generations [194]. However, in the last experiment, a small number of Pareto's solutions have been acquired over the last five generations for heart disease dataset, as shown in Figure 50 based on the results in Table 26

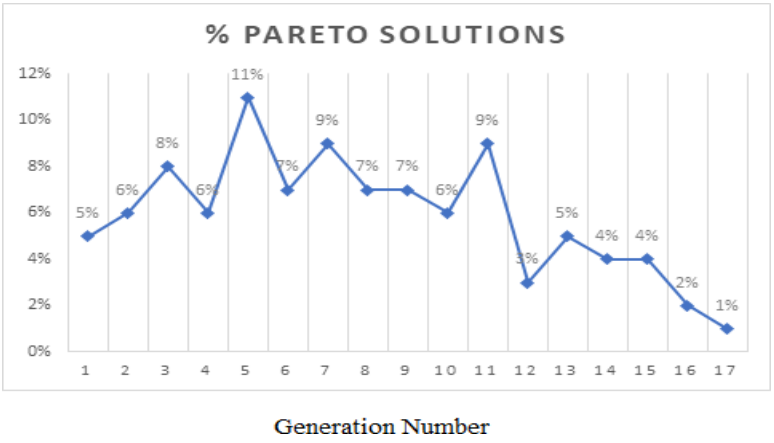


Figure 49: Percentage of solutions in final Pareto set by generation Number

Table 25: Number of Pareto solutions foundat each generation for Heart disease dataset

Generation	Number of Pareto solutions
1	5
2	6
3	8
4	6
5	11
6	7
7	9
8	7
9	7
10	6
11	9
12	3
13	5
14	4
15	4
16	2
17	1
No. of Pareto Solutions	100

In this set of experiments, we searched about the Pareto points on seven datasets. According to the results in Table 26, which presents the characteristics of Pareto solutions for some of the datasets. For Heart disease dataset, solution number 38 outperforms others concerning both criteria. It has an accuracy of 0.9444444, 0.3499654 of the thresholds, and. It is represented as a dark black point in the top of the plot in Figure 48. Five classifiers (numbers 7,37,50,58 and 69) were from the bottom plot in Figure 48 (dark black points). The classifier number 37 for Pima dataset provides the best results, and its accuracy is 0.8275862 and 0.1850489 of the average of thresholds. Also, looking at the results for Indian liver patient data and Thyroid dataset one can notice that it is not worth considering one criterion. In the case of when many Pareto points are obtained, the selection of the classifier becomes a difficult task [1].

Table 26: Pareto solutions for datasets

N	Dataset	Classifier number	ACC	T
1	Heart disease	38	0.9444444	0.3499654
2	Pima dataset	37	0.8275862	0.1850489
3	Breast Cancer dataset	34	0.9911504	0.3495971
4	Indian liver patient data	13	0.65909091	0.07943844
5	Hepatitis dataset	77	0.99	0.6593548
6	Thyroid dataset	69	0.99	0.086025
7	Cardiotocographic dataset	30	0.9814815	0.4218750

For clarity, we choose heart disease dataset to explain how the concept development. However, all the results of all datasets are necessary to compare the outcomes of the experiments.

Table 27 presents the characteristics of some of the Pareto solutions. Solution number that outperforms others concerning both criteria for Pima dataset, Breast Cancer dataset, Cardiotocographic dataset, Hepatitis dataset, Thyroid dataset and Indian Liver dataset. It is represented as a dark red point in the top of the plot in the next Figure. Some classifiers were the worst performance based on both criteria. The rest of the points represent the generations of Pareto point between the best solution and the worse solution

Table 27: Samples of Pareto solutions

N	dataset	Classifier number	ACC	T
1	Pima dataset	37	0.8275862	0.1850489
2	Breast Cancer dataset	34	0.9911504	0.3495971
3	Cardiotocographic dataset	30	0.9814815	0.4218750
4	Hepatitis dataset	1	0.97	0.657419
5	Thyroid dataset	69	0.98	0.086025
6	Indian Liver dataset	13	0.65909091	0.07943844

Figure 50 presents the characteristics of some of the Pareto solutions for Breast Cancer dataset, Pima dataset, Hepatitis dataset and Cardiotocographic dataset. Solution number 34 outperforms others concerning both criteria for Breast Cancer dataset. It has an accuracy of 0.99 and 0.3495971of the thresholds. It is represented as a dark red point in the top of the plot in Figure 50. Five classifiers were the worst performance based on both criteria. They are in the bottom plot in Figure 50 (dark red points). The rest of the points in the plot in Figure 50 for represent the generations of Pareto point between the best solution and the worse solution. Similarly, solution number 37 outperforms others concerning both criteria for Pima dataset. It has an accuracy of 0.8275862 and 0.1850489 of the thresholds. Cardiotocographic dataset has an accuracy of 0.9814815 and thresholds of 0.4218750 for

solution number 30. Solution number 1 outperforms others concerning both criteria for Hepatitis dataset with accuracy of 0.97 and thresholds of 0.657419.

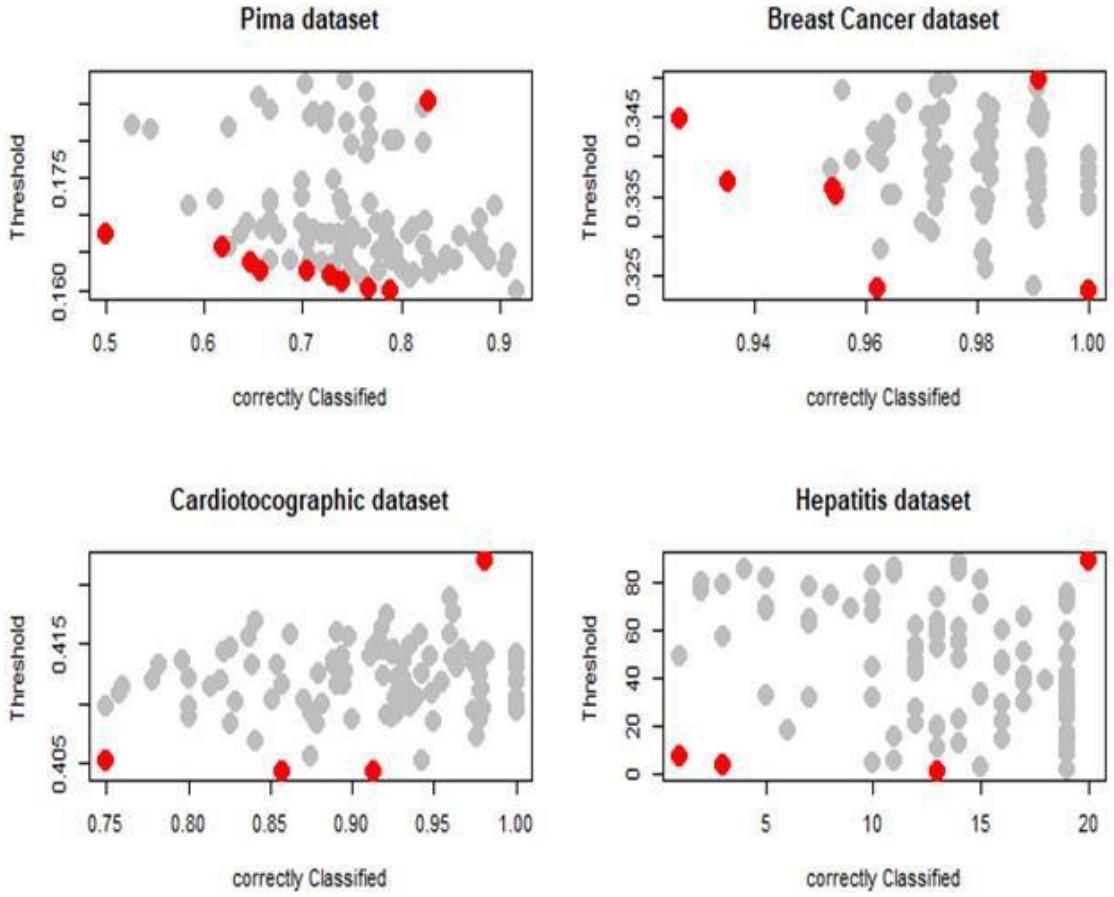


Figure 50: Pareto solutions for datasets, Trade-offs between classification objectives

The characteristics of some of the Pareto solutions for Thyroid dataset and Indian Liver dataset are represented in Figure 51. Solutions 69 and 13 outperform others concerning both criteria for Thyroid dataset and Indian Liver dataset respectively. The best solutions are represented as dark red points in the top of the plot. Classifiers were the worst performance based on both criteria are in the bottom plot (dark red points). The rest of the points in the plot represent the generations of Pareto point between the best solution and the worse solution. The decision-maker can choose based on the purpose of the analysis goal in this stage to obtain more accurate results.

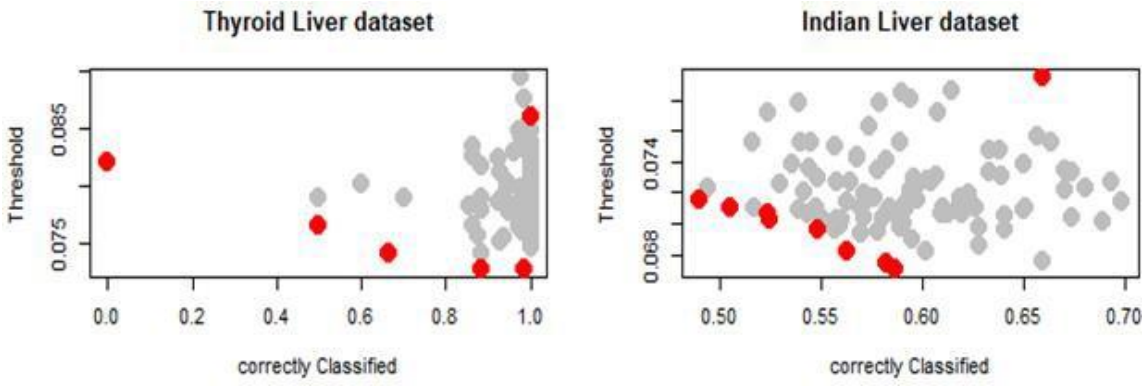


Figure 51: Pareto solutions for datasets, Trade-offs between classification objectives

6.4 Summary

This section aimed to apply the Pareto optimality approach for optimising the accuracy, the data rate involved and the threshold of a classifier. The empirical results generated in Chapter 4 were used to assist in the selection of an appropriate classifier from a collection of classifiers. This approach considers not only the classification accuracy but also the potential trade-offs between classification objectives. However, we do not expect to identify the single classifier that performs best on all data sets following the No Free Lunch theorem [193]. Naturally, by using different classification techniques to enhance the performance, this study could be expanded to consider the ideal set of features by using different feature selection methods.

It is essential to mention that the specialists in any domain may prefer to build machine learning models based on the data collected from the same area, and then use the model to find the best results. Assessing the AD of the model may help to accommodate this preference. However, in the case of diseases, the specialists may know the applicability domain of the model before using it, the model can be reused to obtain the best outcomes.

7 Conclusion and future work

The ML techniques are highly in demand in many fields because of the continuous increase in data sources and limited use of the models, especially in the healthcare field. This research work aimed to solve this problem by proposing an assessing framework for the classification model with a specific focus on where the model is successful and useful. This chapter concludes the overall performance of the proposed framework, along with the contributions. Moreover, the conclusions are drawn from this work. Future directions are also provided in this chapter to enhance this research work. Additionally, this chapter also presents the limitations of this study. Finally, the R program language was utilised to implement this framework.

The scheme of the proposed approach for investigation of the usefulness of the AD approach for ML classification model is illustrated in Figure 52. The implementation and the validation of this approach are performed using three different methods, i.e. 1) assessing the applicability domain of classifiers (ADOC). 2) the robustness of the classification model based on applicability domain approach, 3) a classifier automatically selected using the Pareto set approach.

The main aim of this study is to investigate the connection between the applicability domain approach and the classification model performance. We are examining the usefulness of assessing AD for the classification model, i.e. reliability, reuse, robustness of classifiers. The work is implemented using three approaches, and these approaches are conducted in three various attempts. Firstly, assessing the applicability domain for the classification model. Secondly, investigating the robustness of the classification model based on the applicability domain approach. Thirdly, selecting an optimal model using Pareto optimality. The experiments in this work are illustrated by considering different machine learning algorithms for binary and multi classifications for healthcare datasets from the official benchmark data repository. In the first approach, the decision trees algorithm (DT) is used for the classification of datasets in the classification stage. The feature selection method is applied to choose features for classification. The obtained classifiers are used in the third approach for selection model using Pareto optimality. The second approach is implemented using three steps; namely, building classification model; Generating synthetic data; and evaluating the obtained results.

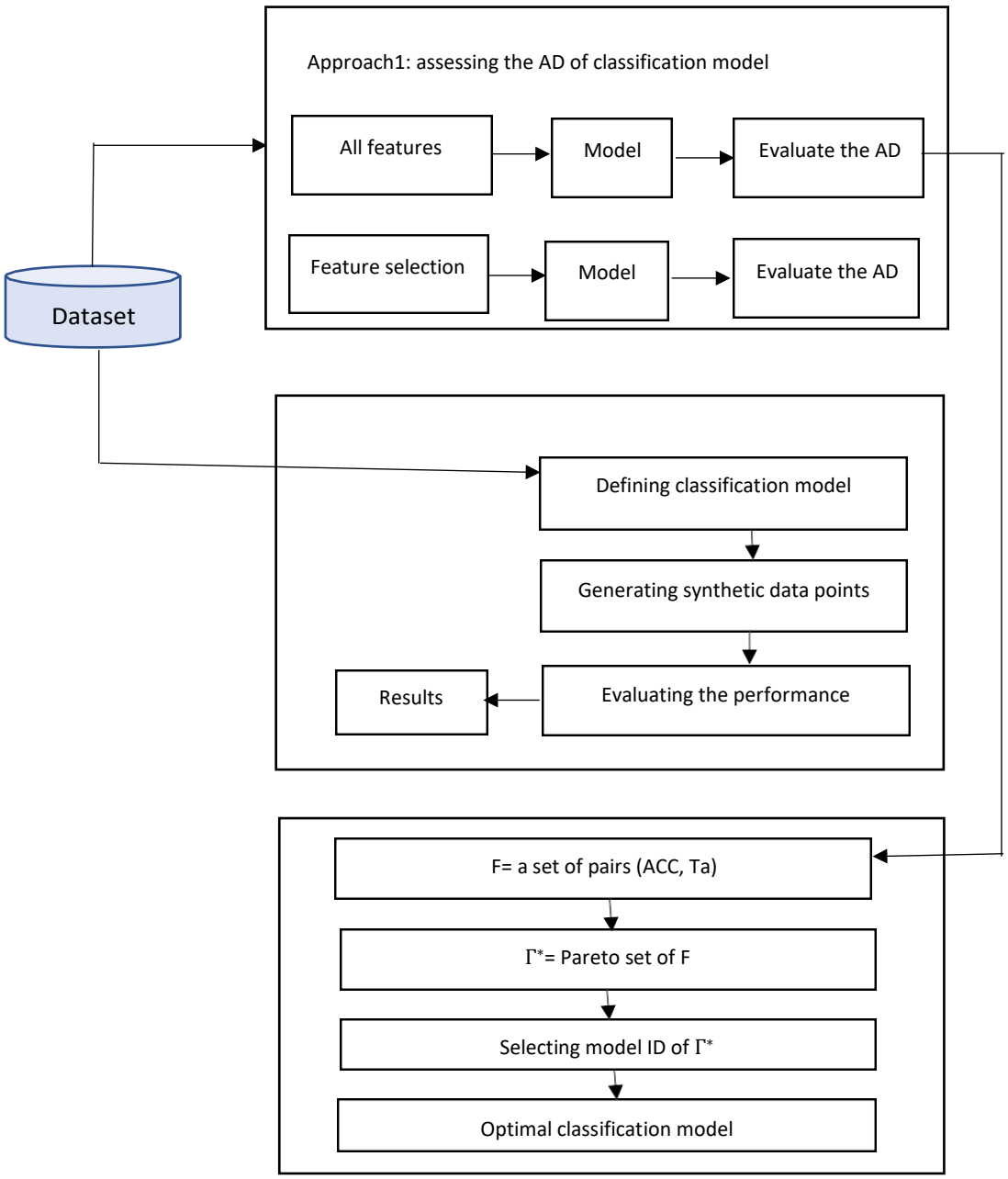


Figure 52: All the approach presented in this work for classification model

7.1 Discussion

It is useful to build machine learning models, but it can be time-consuming work. Construction and reuse of models could be helpful wherever feasible. However, classification models can perform effectively for the assignment they are being trained based on. However, to make the classification model work in new environments, some adaptation may also be needed. In this study, we attempted to provide some ideas on different assignments to assess a model's effectiveness. Besides, the reuse of a classification model in a different environment of a new problem can be achieved.

Every study conducted has research questions as well as targets and objectives. Chapter 1 discussed the questions, goals and objectives of this work. This study targets to propose a framework for investigating the use of AD for the model of classification. Also, explore the challenge of evaluating the classification model's applicability domain by developing a framework for healthcare data. Therefore, the existing state-of-art literature was explored evaluating the classification model and machine learning algorithms for assessing the applicability domain approach to identify the gaps in this domain. The gaps are related to assessing AD of the ML model and investigating the robustness of the classification model considering the applicability domain concept. These challenges, the background, and justification of this research study are explained in Chapter 2

To attempt to define the applicability domain of classifier, and investigate the usefulness of the applicability domain concept is in the evaluation of the classification model is performed. The framework in figure 40 was implemented using three approaches, i.e. investigation of the applicability domain of classification model, Robustness of classification model based on applicability domain approach, and classifier automatically selected using Pareto points approach. The methodology of the framework is discussed in Chapter 3.

Chapter 4 presents the results of an investigation of the applicability domain of classifiers (ADOC) in detail. In this approach, the procedure of the proposed algorithm (ADOC) is described. Firstly, the averages of the neighbourhood width to all instances of the training set are computed. Next, ensemble classifier is constructed by using decision trees classifiers. Then the measures of the bias and the precision of the ensemble classifier outcomes are calculated. Combination of these measures and the averages of the neighbourhood width are used to estimate the reliability associated with the classification model of each data point. In this chapter, we answered the first research question.

Can the Applicability Domain be defined such as the Machine Learning classification model will tolerate a new extended data subset reliably? What will be then the effect of the Feature Selection method choice on the assessment of the Applicability Domain?

In this work, the AD of the classification model is defined as the reliability of the classifier at each data point. We do not consider the nearest data points as in KNN. We compute the average of the distances between each instance and the remain instances from the training set. The neighbourhood width is computed for each data point by computing STD at each data point. Therefore, the AD of the classifier is the mean value of the neighbourhood width of all the training data points. Thus, we assessed the accuracy and the data rate within the AD of the classification model. We determine whether testing

set examples to fall within the training set neighbourhood. The performance of the classifier at each data point is computed as 1 for cases in the test set correctly predicted and 0 otherwise. It is essential to notice that the neighbourhood width at each training instance is determined based on its reliability, which is measured by combining bias and precision. Lastly, optimal features set used in the algorithm to assess the impact of the FS method on the algorithm results.

In chapter 5, robustness of the classification model based on applicability domain approach is discussed. The model trained using random forests classifier. We applied the three phases of this model as

1. The first stage: Defining the classification model.
2. The second stage: Generating synthetic data points
3. The third stage: Evaluating the performance

In this chapter, we answered the second research question How can the robustness of the Machine Learning model be evaluated by considering the Applicability Domain concept in the evaluation of the classification model?

The robustness of the model is evaluated based on the prediction of the model. The model was estimated as the following parameters:

1. Percentage of samples that have good accuracy.
2. Rate of examples that have poor accuracy.

We follow a similar approach in ML such as the weight in the neural network. In the neural network, the weight of the learning process can be started with a small random value and then these values are changed based on the feedback or an optimization process. However, we choose a small number to demonstrate the concept of AD. However, this number should be optimized by following some criteria related to the optimization of a cost function as we mentioned in future work.

The overall implementation of this approach was performed in three different stages that efficiently used the features of the AD concept to define a model's robustness. Significant features this proposed contribution include measuring the distances to identify the close points. Thus, using synthetic data to test the robustness of the model is used. Next, defining the threshold for each test data, and optimise the threshold parameter r. The distance matrix of the training set is considered. Then, the model's response domain to reflect upon the reliability of results derived in its descriptor space. This proposed method identifies an appropriate random value where the model accuracy changes to define the robustness in the model.

The overall performance of this approach is almost similar to the results of the previous method. However, the accuracy and the data rate reduce based on the threshold value.

In the third approach which is discussed in Chapter 6, a classifier automatically selected using the Pareto set approach of a collection of classifiers obtained from the method of assessing the AD of a classifier (in chapter 4). After getting the ensemble classifier from the approach of determining the AD for classifiers in chapter 4, the available collections of models are used for finding a better model among them. The identification of this model is performed based on Pareto optimality, which mines classification model collections and identifies a model offers excellent performance for the test set. We aimed to investigate which of these models can perform better for data that lie in the AD of the model based on multi-objectives problem.

The results obtained from the study provided an understanding of how the proposed approach can help to define the model's robustness and the applicability domain, for providing reliable outputs. These approaches open opportunities for classification data and model management. The proposed algorithms are implemented, tested and validated through a set of experiments — a classification accuracy of instances that fall in the domain of the model. For the first approach, by considering all the features; the highest accuracy obtained is 0.98, with thresholds average of 0.34 for Breast cancer dataset. After applying feature selection method, the accuracy is 0.96% with 0.27 thresholds average. For the robustness of the classification model based on the applicability domain approach, the minimum accuracy is 0.62% for Indian Liver Patient data at r=0.10, and the maximum accuracy is 0.99% for Thyroid dataset at r=0.10. For the selection of an optimal model using Pareto optimality, the optimally selected classifier gives the accuracy of 0.94% with 0.35 thresholds average.

7.2 Contributions

Reusing current classification models can be useful in different fields, including the healthcare field. Because of a significant quantity of accessible information and models used for data analysis, this has become one of the critical problems. The focus of this study is the assessment of the applicability domain of classification models. This section summarises the work described in the thesis that highlights the primary contributions, discusses new open issues and makes suggestions for future work. The developed framework has contributed by offering some advantages. It may assist healthcare specialists in making their decision while classifying a new dataset. Moreover, it can be useful in assessing the AD of the classification models.

The following contributions are accomplished, which can be outlined as follows:

- The existing state-of-art literature was explored evaluating the classification model and machine learning algorithms for assessing the applicability domain approach to identify the gaps in this domain.

- An attempt to define the applicability domain of a classifier was performed. Also, investigating the usefulness of the applicability domain concept for the classification model was achieved by using the following three approaches:

  a) Investigation of the applicability domain of classification model.

  b) Robustness of classification model based on applicability domain approach.

  c) Classifier automatically selected using Pareto points method.

The measures of the accuracy, error rate and data rate in the AD are considered for the classifier.

- The applicability domain of classifiers (ADOC) approach. This approach is provided in chapter4. Parts of this chapter is presented in a poster (under review) in annual innovative engineering research conference (AIERC 2019), University of Bradford,31,08,2019. Firstly, we compared the results of the ensemble classifier in term of correctly classified instances (accuracy) on the datasets. The classifier provided the highest average accuracy of 99.65 % on Thyroid dataset. However, the accuracy was only 69.23% on Indian Liver Patient data set. Secondly, we computed the ACC of the data that fall within the AD of the model. T is calculated as well for the model based on averaging all the threshold obtained for each instance. When we consider the accuracy measure, the classifier provides the highest accuracy of 0.9771833 on Breast-cancer dataset. The thresholds average is 0.3409341. However, when we look at the highest value of thresholds averaging, the classifier provides 0.6416129 on Hepatitis dataset, the accuracy on the Hepatitis dataset is 0.9086933. After selecting some features by using RFE method, the classifier provided the highest accuracy of 0.9609411 on Breast-cancer dataset, and the threshold average was 0.2704212. However, the classifier provided 0.3522581 on Hepatitis dataset, and the accuracy on the Hepatitis dataset is 0.8399648.

- The robustness of classification model based on applicability domain approach. This approach is provided in chapter5. This chapter is presented as a research paper (under review) in the journal of Expert system, Wiley,22,07,2019. Firstly, Evaluating the performance of the RF classifier on five data sets is given. The RF algorithm has the highest accuracy on Thyroid dataset. However, at the liver data set, Random Forests provides the lower accuracy. We consider the rate of points that fall in the AD of the model. The value of r has the sequence values between 0.10 and 0.20 by increment 0.10. Each time we add r to the training set, we

test the points from the test set that fall in the same area (within the AD) with the training set. Applying the classifier on only these points yields the following results: Rf classifier gave the highest number of correctly classified on Thyroid dataset for all values of r (0.10 into 0.20). The accuracy is 0.99. The first value of r is 0.10; the points number is 124. The model classified 123 points correctly. However, the last value of r is 0.20; RF classified 74 points correctly from 75 points in total. The accuracy is low of correctly classified on Indian Liver Patient data for all values of r (0.10 into 0.20). The first value of r is 0.10; the points number is 62. the model classified 32 points correctly with 62 of accuracy. However, the last value of r is 0.20; RF classified 10 points correctly from 15 points in total with an accuracy of 67.

- The classifier automatically selected using Pareto points approach. This approach is provided in chapter6. This chapter is presented as a short research paper (accepted paper) in AI-2019 Thirty-ninth SGAI international conference on artificial intelligence. Cambridge, England, 17-19 December 2019. This section aimed to apply the Pareto optimality approach for optimising the accuracy, the data rate involved and the threshold of a classifier. After getting the ensemble classifiers from the method of assessing the AD for classifiers in chapter 4, the available collections of models are used for finding a better model among them. All objectives are evaluated and sorted to find the Pareto set of optimal solutions from the feasible solutions based on the values of accuracy (ACC) and Thresholds average (Ta) of each model. According to the results, Solution number 38 outperforms the others concerning both criteria for Heart disease dataset. It has an accuracy of 0.9444444, 0.3499654 of the thresholds. However, five classifiers (number 7,37,50,58 and 69) were from the bottom plot in Figure49 (dark black points). The rest of the points represent the generations of Pareto point between the best solution and the worse solution.

- Concluding this work introduced the concept of applicability domain for classifiers and tested the use of this concept with some case study.

7.3 Limitations

Each research does have some limitations. Although these approaches have been successfully applied in some benchmark datasets, which are available publicly, the limitation of this study is the use of small datasets.

Using binary classifiers is usually faster. However, we need to try using datasets with many classes considering some problems associated with multi-class (i.e., class imbalances that introduce bias).

As far as we read, most of the studies of the evaluation of the performance of ML algorithms focus on how to improve the classification performance, especially the accuracy, while the robustness is not taken into consideration.

7.4 Future work

These approaches can be applied in a wide range of domains such as education or economy. Using different ML techniques to build classifiers may affect the results. Uncertainty can derive from many sources including incomplete observability and incomplete modelling. The most important question for us was how the different values of threshold r can affect the outcomes.

Further work that could be conducted, as a result of these findings, this study would provide an information for future research to boost the results, thus, the proposed ADOC algorithm can be improved using meta-metrics (as shown in section 2.3.5). The objective of introducing meta-metrics is to make the model robust enough to distinguish between the new data (in AD or out AD). Additional datasets, including a big data set, can be added to the implemented algorithm. We are only classifying

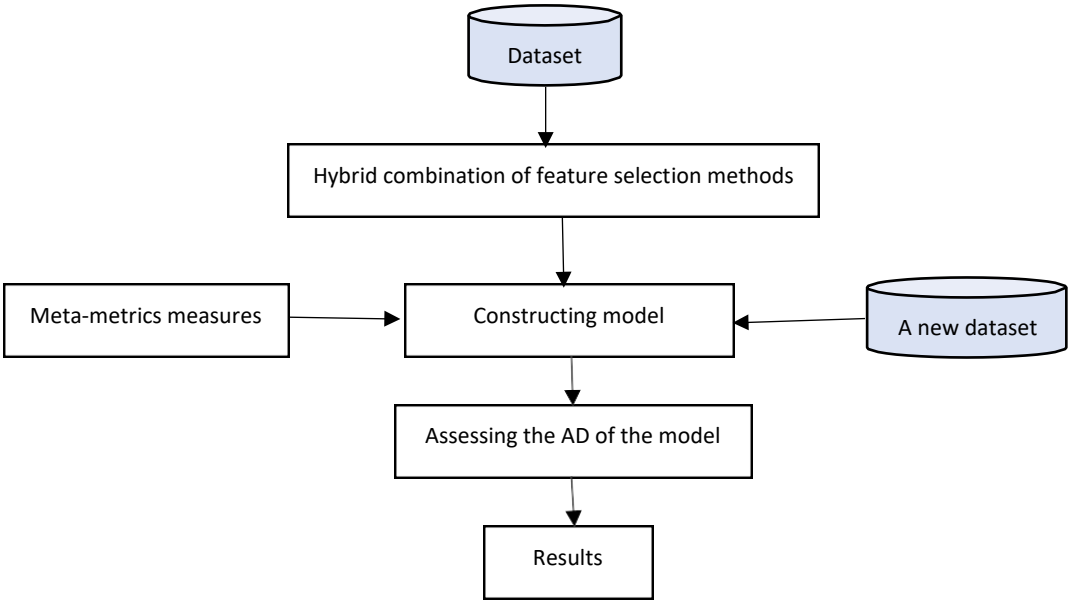


Figure 53: Extension of the proposed algorithm

seven datasets in this study. However, by adding more dataset such dataset from the healthcare domain, it can be easily expanded. In Figure 53, the extension of the proposed algorithm could be an abstract level.

Our results indicated that better and more efficient assessing the AD is one important factor, which might be useful for better performance of the evaluation task of the classifiers.

reusing a model in a new environment.

Another interesting extension to the research would be to apply these approaches on much larger data sets which display a wider variety of class distributions.

The effect of feature selection methods on analyses high-dimensional heterogeneous healthcare data sets can be explored using different feature selection methods and various classification techniques. Furthermore, the impact of hybrid combinations of features selection approaches can be investigated and evaluated on big healthcare datasets. In our future work, we will investigate the impact of other feature selection approaches as well as hybrid feature selection techniques to get the optimal set of features. Our approach focuses primarily on classification models, and we intend to extend this strategy to regression models in the future, including the assessment of model features in data space partitioning. A new exciting direction might be used to estimate the model reliability for a new dataset. In Chapter 5, for simplicity, the r-value in this study is for obtaining the best accuracy. Here we choose a small number to demonstrate the concept of the AD. In future work, we are going to look at how this number can be generated to optimize the cost function. In the next step, we are going to review the related work on Big Data associated studies and the AD techniques from the machine learning literature.

Following the study in [31], the classifier was trained on the merged databases under different conditions and this gives promisingly robust classification results. Future works will be carried out with assessing the AD of merged databases.

7.5 Summary

From the papers reviewed and discussed, the data mining methods performance including the accuracy varies depending on some factors such as the features of the data sets, the sample size of the available data, and the size of data set between the training and testing sets. The characteristic properties among the healthcare data sets are imbalanced data sets, where the majority and the minority classifier are not balanced. Thus, the performance becomes poor when run by the classifiers. Moreover, another characteristic of the healthcare data set is the missing values. There is no one appropriate data mining approach to solve all these issues. Following the review of some previous work for evaluating of ML models as stated above, most of the existing research conducted in the area focused less on AD approach. However, they look more into measurements such as the accuracy measure. They also applied the standard technique of evaluation into training and testing processes.

Our proposed approaches did not only use common evaluation measurements but utilising the AD approach. Moreover, trying to assess the AD of the classifiers based on each instance in the dataset and proposed method. They are also focused mainly on determining AD related to the classification model. This chapter addressed the discussion of the achieved results, the chosen methodology, and the validity of the experiments. Construction and then reuse of models will make sense where possible. However, the models can perform well in the task they are being trained on and that some adaptation may be needed to help to work in new problem space.

Finally, this work has provided a background of the scientific aspects of machine learning and the applicability domain. It is essential to understand the significant elements of machine learning evaluation.

Several measurements of the evaluation have been discussed that can evaluate classification models. Although the current methods which provide AD assessment are limited, they are useful in achieving good results with QSAR models. This research investigates critical aspects of the applicability domain as related to the robustness of classification machine learning algorithms. However, the performance of machine learning techniques depends on the degree of reliable predictions of the model. In the literature, the robustness of the machine learning model can be defined as the ability of the model to give the testing error is close to the training error. Moreover, it is the properties that describe the stability of the model performance when being tested on the new datasets. Concluding this thesis introduced the concept of applicability domain for classifiers and tested the use of this concept with some case study.

References

[1] A. Palczewska, D. Neagu, and M. Ridley, "Using pareto points for model identification in predictive toxicology," *J. Cheminform.*, vol. 5, no. 3, pp. 1–16, 2013.

[2] Y. Jin, S. Member, B. Sendhoff, and S. Member, "Pareto-Based Multiobjective Machine Learning: An Overview and Case Studies," *IEEE Trans. Syst. Man, Cybern. Part C (Applications Rev.*, vol. 38, no. 3, pp. 397–415, 2008.

[3] L. Breiman, "Randon Forests," pp. 1–35, 1999.

[4] F. Sahigara, "Defining the Applicability Domain of QSAR models: An overview."

[5] N. Fechner, A. Jahn, G. Hinselmann, and A. Zell, "Estimation of the applicability domain of kernel-based machine learning models for virtual screening," *J. Cheminform.*, vol. 2, no. 1, pp. 1–20, 2010.

[6] M. Mohammed, M. B. Khan, and E. B. M. Bashier, *Machine learning : algorithms and applications*. .

[7] J. Han and M. Kamber, *Data mining : concepts and techniques*. Elsevier, 2006.

[8] A. Malaviya, *Data Mining and Analysis.* 2014.

[9] Y. Zhao, *R and data mining : examples and case studies*. Academic Press, 2013.

[10] M. H. Dunham, *Data mining introductory and advanced topics*. Prentice Hall/Pearson Education, 2003.

[11] K. Meinke and A. Bennaceur, "Machine learning for software engineering," in *Proceedings of the 40th International Conference on Software Engineering Companion Proceeedings - ICSE '18*, 2018, pp. 548–549.

[12] S. Rogers and M. Girolami, *A first course in machine learning*. .

[13] K. P. Murphy, *Machine learning : a probabilistic perspective*. MIT Press, 2012.

[14] J. Ledolter, *Data mining and business analytics with R*. .

[15] E. Alpaydın, "Introduction to machine learning," *Methods Mol. Biol.*, vol. 1107, pp. 105–128, 2014.

[16] H. B. Barlow, “Unsupervised Learning,” *Neural Comput.*, vol. 1, no. 3, pp. 295–311, Sep. 1989.

[17] J. (Computer scientist) Bell, *Machine learning : hands-on for developers and technical professionals*. .

[18] C. C. Aggarwal, *Data classification : algorithms and applications*. .

[19] X. Wu and V. Kumar, *The top ten algorithms in data mining*. CRC Press, 2009.

[20] C. Chio and D. Freeman, *Machine learning and security : protecting systems with data and algorithms*. .

[21] W. F. Schneider and H. Guo, “Machine Learning,” *J. Phys. Chem. A*, vol. 122, no. 4, pp. 879–879, Feb. 2018.

[22] P. Ahmad, S. Qamar, and S. Qasim Afser Rizvi, “Techniques of Data Mining In Healthcare: A Review,” *Int. J. Comput. Appl.*, vol. 120, no. 15, pp. 38–50, 2015.

[23] M. Crispino, “Machine Learning Algorithms and Techniques,” *GeneXus*, no. January, 2018.

[24] P. A. Flach, *Machine learning : the art and science of algorithms that make sense of data*. Cambridge University Press, 2012.

[25] N. Lavesson, V. Boeva, E. Tsiporkova, and P. Davidsson, “A method for evaluation of learning components,” *Autom. Softw. Eng.*, vol. 21, no. 1, pp. 41–63, 2014.

[26] P. Kaur, M. Sharma, and M. Mittal, “Big Data and Machine Learning Based Secure Healthcare Framework,” *Procedia Comput. Sci.*, vol. 132, pp. 1049–1059, 2018.

[27] J. Han, M. Kamber, and J. Pei, *Data mining : concepts and techniques*. Elsevier Science, 2011.

[28] M. J. Zaki Wagner Meira Jr, “Data Mining and Analysis: Fundamental Concepts and Algorithms.”

[29] X. (Jerry) Zhu, “Semi-Supervised Learning Literature Survey,” 2005.

[30] O. Chapelle, B. Scholkopf, and A. Zien, Eds., “Semi-Supervised Learning (Chapelle, O. et al., Eds.; 2006) [Book reviews],” *IEEE Trans. Neural Networks*, vol. 20, no. 3, pp. 542–542, Mar. 2009.

[31] M. Shami and W. Verhelst, “An evaluation of the robustness of existing supervised machine learning approaches to the classification of emotions in speech,” *Speech Commun.*, vol. 49, no. 3, pp. 201–212, 2007.

[32] D. Hand, D. Hand, H. Mannila, H. Mannila, P. Smyth, and P. Smyth, *Principles of data mining*, vol. 30. 2001.

[33] M. A. and R. Khanna, *Efficient learning machines : theories, concepts, and applications for engineers and system designers*. [New York, NY] : Apress Open, [2015]., 2015.

[34] G. J. Myatt and W. P. Johnson, *Making sense of data III : a practical guide to designing interactive data visualizations*. Wiley, 2012.

[35] X. Zhu, S. Zhang, Z. Jin, Z. Zhang, and Z. Xu, "Missing value estimation for mixed-attribute data sets," *IEEE Trans. Knowl. Data Eng.*, vol. 23, no. 1, pp. 110–121, 2011.

[36] O. Rado, M. Al Fanah, and E. Taktek, "Performance Analysis of Missing Values Imputation Methods Using Machine Learning Techniques," Springer, Cham, 2019, pp. 738–750.

[37] A. Ghosh, N. Manwani, and P. S. Sastry, "On the robustness of decision tree learning under label noise," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 10234 LNAI, pp. 685–697, 2017.

[38] A. Hines, P. Kendrick, A. Barri, M. Narwaria, and J. A. Redi, "ROBUSTNESS AND PREDICTION ACCURACY OF MACHINE LEARNING FOR OBJECTIVE VISUAL QUALITY ASSESSMENT e de Nantes , France Vrije Universiteit Brussel and iMinds , Belgium $ Delft University of Technology , the Netherlands," no. Ml.

[39] "Introduction to Data Mining," in *Discovering Knowledge in Data*, Hoboken, NJ, USA: John Wiley & Sons, Inc., 2005, pp. 1–26.

[40] T. Kanamori, S. Fujiwara, and A. Takeda, "Robustness of learning algorithms using hinge loss with outlier indicators," *Neural Networks*, vol. 94, pp. 173–191, 2017.

[41] I. Brown and C. Mues, "An experimental comparison of classification algorithms for imbalanced credit scoring data sets," *Expert Syst. Appl.*, vol. 39, no. 3, pp. 3446–3453, Feb. 2012.

[42] X. Wu *et al.*, "Top 10 algorithms in data mining," *Knowl Inf Syst*, vol. 14, pp. 1–37, 2008.

[43] "Feature selection in machine learning: A new perspective," *Neurocomputing*, vol. 300, pp. 70–79, Jul. 2018.

[44] R. Genuer, J. M. Poggi, and C. Tuleau-Malot, "Variable selection using random forests," *Pattern Recognit. Lett.*, vol. 31, no. 14, pp. 2225–2236, 2010.

[45] G. Chandrashekar and F. Sahin, “A survey on feature selection methods q,” 2013.

[46] L. Yin, Y. Ge, K. Xiao, X. Wang, and X. Quan, “Feature selection for high-dimensional imbalanced data,” *Neurocomputing*, vol. 105, pp. 3–11, Apr. 2013.

[47] P. Hajek and K. Michalak, “Feature selection in corporate credit rating prediction,” *Knowledge-Based Syst.*, vol. 51, no. 1, pp. 72–84, Oct. 2013.

[48] R. C. Deo, “Machine Learning in Medicine.,” *Circulation*, vol. 132, no. 20, pp. 1920–30, Nov. 2015.

[49] “Deep Learning.” [Online]. Available: http://www.deeplearningbook.org/. [Accessed: 28-Dec-2018].

[50] C. Krieger, “Neural Networks in Data Mining ©1996,” pp. 1449–1453, 1996.

[51] N. Oscar, P. A. Fox, R. Croucher, R. Wernick, J. Keune, and K. Hooker, “Machine learning, sentiment analysis, and tweets: An examination of Alzheimer’s disease stigma on Twitter,” *Journals Gerontol. - Ser. B Psychol. Sci. Soc. Sci.*, vol. 72, no. 5, pp. 742–751, 2017.

[52] A. Shenfield and S. Rostami, “A multi objective approach to evolving artificial neural networks for coronary heart disease classification,” *2015 IEEE Conf. Comput. Intell. Bioinforma. Comput. Biol. CIBCB 2015*, 2015.

[53] K. Kourou, T. P. Exarchos, K. P. Exarchos, M. V. Karamouzis, and D. I. Fotiadis, “Machine learning applications in cancer prognosis and prediction,” *Comput. Struct. Biotechnol. J.*, vol. 13, pp. 8–17, 2015.

[54] D. T. Larose, *Data mining methods and models*. .

[55] S. E. Buttrey, “Data Mining Algorithms Explained Using *R*,” *J. Stat. Softw.*, vol. 66, no. Book Review 2, Aug. 2015.

[56] M. C. Keerrthega and D. Thenmozhi, “Identifying Disease -Treatment Relations Using Machine Learning Approach,” *Procedia Comput. Sci.*, vol. 87, pp. 306–315, Jan. 2016.

[57] G. Aruni, G. Amit, and P. Dasgupta, “New surgical robots on the horizon and the potential role of artificial intelligence,” *Investig. Clin. Urol.*, vol. 59, no. 4, pp. 221–222, 2018.

[58] N. Jothi, N. A. Rashid, and W. Husain, “Data Mining in Healthcare - A Review,” *Procedia Comput. Sci.*, vol. 72, pp. 306–313, 2015.

[59] G. Manogaran and D. Lopez, “A survey of big data architectures and machine learning

algorithms in healthcare," *Int. J. Biomed. Eng. Technol.*, vol. 25, no. 2/3/4, p. 182, 2017.

[60] N. Sharma and H. Om, "Data mining models for predicting oral cancer survivability," *Netw. Model. Anal. Heal. Informatics Bioinforma.*, vol. 2, no. 4, pp. 285–295, 2013.

[61] A. Abdelaziz, M. Elhoseny, A. S. Salama, and A. M. Riad, "A machine learning model for improving healthcare services on cloud computing environment," *Meas. J. Int. Meas. Confed.*, vol. 119, no. January, pp. 117–128, 2018.

[62] R. Armañanzas, C. Bielza, K. R. Chaudhuri, P. Martinez-Martin, and P. Larrañaga, "Unveiling relevant non-motor Parkinson's disease severity symptoms using a machine learning approach," *Artif. Intell. Med.*, vol. 58, no. 3, pp. 195–202, 2013.

[63] C. H. Jen, C. C. Wang, B. C. Jiang, Y. H. Chu, and M. S. Chen, "Application of classification techniques on development an early-warning system for chronic illnesses," *Expert Syst. Appl.*, vol. 39, no. 10, pp. 8852–8858, 2012.

[64] S. Fei, "Diagnostic study on arrhythmia cordis based on particle swarm optimization-based support vector machine," *Expert Syst. Appl.*, vol. 37, no. 10, pp. 6748–6752, Oct. 2010.

[65] B. Zheng, S. W. Yoon, and S. S. Lam, "Breast cancer diagnosis based on feature extraction using a hybrid of K-means and support vector machine algorithms," *Expert Syst. Appl.*, vol. 41, no. 4, pp. 1476–1482, Mar. 2014.

[66] M. K. Gupta, K. Agarwal, N. Prakash, D. B. Singh, and K. Misra, "Prediction of miRNA in HIV-1 genome and its targets through artificial neural network: A bioinformatics approach," *Netw. Model. Anal. Heal. Informatics Bioinforma.*, vol. 1, no. 4, pp. 141–151, 2012.

[67] O. Rado, N. Ali, H. M. Sani, A. Idris, and D. Neagu, "Performance Analysis of Feature Selection Methods for Classification of Healthcare Datasets," 2019, pp. 929–938.

[68] R. Miotto, F. Wang, S. Wang, X. Jiang, and J. T. Dudley, "Deep learning for healthcare: review, opportunities and challenges," *Brief. Bioinform.*, vol. 19, no. 6, pp. 1236–1246, Nov. 2018.

[69] O. Faust, Y. Hagiwara, T. J. Hong, O. S. Lih, and U. R. Acharya, "Deep learning for healthcare applications based on physiological signals: A review," *Comput. Methods Programs Biomed.*, vol. 161, pp. 1–13, Jul. 2018.

[70] *International journal of computer science and information security.* IJCSIS Publication.

[71] A. Jutel and D. Lupton, "Digitizing diagnosis: a review of mobile applications in the diagnostic process," *Diagnosis*, vol. 2, no. 2, pp. 89–96, Jun. 2015.

[72] S. R. Bhagya Shree and H. S. Sheshadri, “Diagnosis of Alzheimer’s disease using Naive Bayesian Classifier,” *Neural Comput. Appl.*, vol. 29, no. 1, pp. 123–132, Jan. 2018.

[73] “Cardiovascular Disease Statistics 2017 - heart statistics - BHF.” [Online]. Available: https://www.bhf.org.uk/what-we-do/our-research/heart-statistics/heart-statistics-publications/cardiovascular-disease-statistics-2017. [Accessed: 11-Sep-2019].

[74] “WHO | Social determinants of health,” *WHO*, 2018.

[75] K. Y. Ngiam and I. W. Khor, “Big data and machine learning algorithms for health-care delivery,” *Lancet Oncol.*, vol. 20, no. 5, pp. e262–e273, May 2019.

[76] G. Dong and J. Bailey, *Contrast data mining : concepts, algorithms, and applications*. CRC Press, 2013.

[77] A. Singh, N. Thakur, and A. Sharma, “A review of supervised machine learning algorithms,” *2016 3rd Int. Conf. Comput. Sustain. Glob. Dev.*, pp. 1310–1315, 2016.

[78] S. Mohan, C. Thirumalai, and G. Srivastava, “Effective Heart Disease Prediction Using Hybrid Machine Learning Techniques,” *IEEE Access*, vol. 7, pp. 81542–81554, 2019.

[79] “Prediction of fatty liver disease using machine learning algorithms,” *Comput. Methods Programs Biomed.*, vol. 170, pp. 23–29, Mar. 2019.

[80] S. A. Mostafa, A. Mustapha, S. H. Khaleefah, M. S. Ahmad, and M. A. Mohammed, “Evaluating the Performance of Three Classification Methods in Diagnosis of Parkinson’s Disease,” 2018, pp. 43–52.

[81] A. Sivasakthivel, G. T. Shrivakshan, and G. T. Shrivakshan, “A Comparative Study of Diagnosing Thyroid Diseases Using Classification Algorithm,” *Int. J. Adv. Res. Comput. Sci. Softw. Eng.*, vol. 7, no. 8, p. 181, Aug. 2017.

[82] J. R. (John R. Quinlan and J. Ross, *C4.5 : programs for machine learning*. Morgan Kaufmann Publishers, 1993.

[83] H. Kaneko, “Discussion on Regression Methods Based on Ensemble Learning and Applicability Domains of Linear Submodels,” *J. Chem. Inf. Model.*, vol. 58, no. 2, pp. 480–489, 2018.

[84] M. Kantardzic, *Data mining : concepts, models, methods, and algorithms*. .

[85] W. W. Piegorsch, *Statistical data analytics : foundations for data mining, informatics, and knowledge discovery, Solutions manual*. .

[86] S. PANG and J. GONG, “C5.0 Classification Algorithm and Application on Individual Credit Evaluation of Banks,” *Syst. Eng. - Theory Pract.*, vol. 29, no. 12, pp. 94–104, Dec. 2009.

[87] D. McSherry, “Strategic Induction of Decision Trees BT - Research and Development in Expert Systems XV,” *Res. Dev. Expert Syst. XV*, vol. 1, no. Chapter 2, pp. 15–26, 1999.

[88] M. Mathea, W. Klingspohn, and K. Baumann, “Chemoinformatic Classification Methods and their Applicability Domain,” *Mol. Inform.*, vol. 35, no. 5, pp. 160–180, 2016.

[89] A. Liaw and M. Wiener, “Classification and Regression by RandomForest,” 2001.

[90] C. M. Bishop, “Pattern Recognition and Machine Learning Springer Mathematical notation Ni,” *Springer-Verlag New York, Inc.,, Secaucus, NJ, USA*, 2006.

[91] D. T. Larose and C. D. Larose, *Discovering knowledge in data : an introduction to data mining*. .

[92] Z. Ma and A. Kaban, “K-Nearest-Neighbours with a novel similarity measure for intrusion detection,” *2013 13th UK Work. Comput. Intell. UKCI 2013*, pp. 266–271, 2013.

[93] G. F. Fan, Y. H. Guo, J. M. Zheng, and W. C. Hong, “Application of the weighted k-nearest neighbor algorithm for short-term load forecasting,” *Energies*, vol. 12, no. 5, 2019.

[94] T. Elomaa, H. Mannila, and H. Toivonen, *Machine learning : ECML 2002 : 13th European Conference on Machine Learning, Helsinki, Finland, August 19-23, 2002 : proceedings*. Springer, 2002.

[95] M. R. Chernick and R. A. LaBudde, *An introduction to bootstrap methods with applications to R*. Wiley, 2011.

[96] I. H. (Ian H. . Witten, E. Frank, and M. A. (Mark A. Hall, *Data mining : practical machine learning tools and techniques*. Morgan Kaufmann, 2011.

[97] Y.-Q. Wang, “An Analysis of the Viola-Jones Face Detection Algorithm,” *Image Process. Line*, vol. 4, pp. 128–148, Jun. 2014.

[98] V. Consonni, D. Ballabio, and R. Todeschini, “Evaluation of model predictive ability by external validation techniques,” *J. Chemom.*, vol. 24, no. 3–4, pp. 194–201, 2010.

[99] M. Sokolova and G. Lapalme, “A systematic analysis of performance measures for classification tasks,” *Inf. Process. Manag.*, vol. 45, no. 4, pp. 427–437, 2009.

[100] P. Mills, “Efficient statistical classification of satellite measurements,” *Int. J. Remote Sens.*,

vol. 32, no. 21, pp. 6109–6132, 2011.

[101] W. Oude Nijeweme-d'Hollosy, L. van Velsen, M. Poel, C. G. M. Groothuis-Oudshoorn, R. Soer, and H. Hermens, "Evaluation of three machine learning models for self-referral decision support on low back pain in primary care," *Int. J. Med. Inform.*, vol. 110, no. August 2017, pp. 31–41, 2018.

[102] S. Burger, *Introduction to machine learning with R : rigorous mathematical analysis*. .

[103] R. Ali, S. Lee, and T. C. Chung, "Accurate multi-criteria decision making methodology for recommending machine learning algorithm," *Expert Syst. Appl.*, vol. 71, pp. 257–278, 2017.

[104] B. R. Gaines, "Transforming Rules and Trees into Comprehensible Knowledge Structures," *KDD 1996*, 1996.

[105] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognit. Lett.*, vol. 27, no. 8, pp. 861–874, Jun. 2006.

[106] A. P. Engelbrecht, *Fundamentals of computational swarm intelligence*. Wiley, 2005.

[107] O. O. Bittencourt, F. Morelli, C. A. dos Santos Júnior, and R. Santos, "Evaluating Classification Models in a Burned Areas' Detection Approach," Springer, Cham, 2019, pp. 577–591.

[108] "Amazon ditched AI recruiting tool that favored men for technical jobs | Technology | The Guardian." [Online]. Available: https://www.theguardian.com/technology/2018/oct/10/amazon-hiring-ai-gender-bias-recruiting-engine. [Accessed: 22-Oct-2018].

[109] C. Pelletier, S. Valero, J. Inglada, N. Champion, and G. Dedieu, "Assessing the robustness of Random Forests to map land cover with high resolution satellite image time series over large areas," *Remote Sens. Environ.*, vol. 187, pp. 156–168, 2016.

[110] Z. Liu, R. Wang, N. Japkowicz, Y. Cai, D. Tang, and X. Cai, "Mobile app traffic flow feature extraction and selection for improving classification robustness," *J. Netw. Comput. Appl.*, vol. 125, no. February 2018, pp. 190–208, 2019.

[111] N. Elavarasan and K. Mani, "A Survey on Feature Extraction Techniques," *Int. J. Innov. Res. Comput. Commun. Eng. (An ISO*, vol. 3297, no. 1, 2007.

[112] A. Hines, P. Kendrick, A. Barri, M. Narwaria, and J. A. Redi, "ROBUSTNESS AND PREDICTION ACCURACY OF MACHINE LEARNING FOR OBJECTIVE VISUAL QUALITY ASSESSMENT e de Nantes , France Vrije Universiteit Brussel and iMinds , Belgium $ Delft University of

Technology , the Netherlands," *2014 22nd Eur. Signal Process. Conf.*, no. Ml, pp. 2130–2134.

[113] T. I. Netzeva *et al.*, "Current Status of Methods for Defining the Applicability Domain of (Quantitative) Structure-Activity Relationships," *Altern. to Lab. Anim.*, vol. 33, no. 2, pp. 155–173, Apr. 2005.

[114] F. Sahigara, K. Mansouri, D. Ballabio, A. Mauri, V. Consonni, and R. Todeschini, "Comparison of different approaches to define the applicability domain of QSAR models," *Molecules*, vol. 17, no. 5, pp. 4791–4810, 2012.

[115] N. Aniceto, A. A. Freitas, A. Bender, and T. Ghafourian, "A novel applicability domain technique for mapping predictive reliability across the chemical space of a QSAR: Reliability-density neighbourhood," *J. Cheminform.*, vol. 8, no. 1, pp. 1–20, 2016.

[116] A. Dubey, "Machine Learning Approaches in Drug Development of HIV / AIDS," vol. 3, no. 1, pp. 1–4, 2018.

[117] A. Lavecchia, "Machine-learning approaches in drug discovery: Methods and applications," *Drug Discov. Today*, vol. 20, no. 3, pp. 318–331, 2015.

[118] W. Klingspohn, M. Mathea, A. Ter Laak, N. Heinrich, and K. Baumann, "Efficiency of different measures for defining the applicability domain of classification models," *J. Cheminform.*, vol. 9, no. 1, pp. 1–17, 2017.

[119] R. W. Stanforth, E. Kolossov, and B. Mirkin, "A measure of domain of applicability for QSAR modelling based on intelligent K-means clustering," *QSAR Comb. Sci.*, vol. 26, no. 7, pp. 837–844, 2007.

[120] I. Sushko, "Applicability domain of QSAR models: status quo and perspec9ves," 2012.

[121] T. S. Schroeter *et al.*, "Estimating the domain of applicability for machine learning QSAR models: A study on aqueous solubility of drug discovery molecules," *J. Comput. Aided. Mol. Des.*, vol. 21, no. 12, pp. 651–664, 2007.

[122] A. Tropsha, "Predictive Quantitative Structure–Activity Relationship Modeling," *Compr. Med. Chem. II*, pp. 149–165, Jan. 2007.

[123] T. Puzyn, J. Leszczynski, and M. T. Cronin, Eds., *Recent Advances in QSAR Studies*. Dordrecht: Springer Netherlands, 2010.

[124] K. Roy, S. Kar, and P. Ambure, "On a simple approach for determining applicability domain of QSAR models," *Chemom. Intell. Lab. Syst.*, vol. 145, pp. 22–29, 2015.

[125] C. Knauer and K. Kriegel, “On the Bounding Boxes Obtained by Principal Component Analysis,” *2nd Eur. Work. Comput. Geom.*, pp. 193–196, 2006.

[126] S. Boyd and L. Vandenberghe, *Convex Optimization*. .

[127] C. Hull, “Convex Hull,” *Encycl. Neurosci.*, no. 1, pp. 885–885, 2008.

[128] J. Jaworska, N. Nikolova-Jeliazkova, and T. Aldenberg, “QSAR Applicability Domain Estimation by Projection of the Training Set in Descriptor Space: A Review.”

[129] F. Sahigara, K. Mansouri, D. Ballabio, A. Mauri, V. Consonni, and R. Todeschini, “molecules Comparison of Different Approaches to Define the Applicability Domain of QSAR Models,” *Molecules*, vol. 17, pp. 4791–4810, 2012.

[130] “A new procedure for identifying the frame of the convex hull of a finite collection of points in multidimensional space,” *Eur. J. Oper. Res.*, vol. 92, no. 2, pp. 352–367, Jul. 1996.

[131] P. M. Pardalos, “Linear P r o g r a m m i n g A p p r o a c h e s to the C o n v e x Hull P r o b l e m in R m,” vol. 29, no. 7, pp. 23–29, 1995.

[132] G. Fung, R. Rosales, and B. Krishnapuram, “Learning rankings via convex hull separation,” *Adv. Neural Inf. Process. Syst.*, vol. 18, p. 395, 2006.

[133] Y. Kim and J. Kim, “Convex Hull Ensemble Machine for Regression and Classification,” *Knowl. Inf. Syst.*, vol. 6, no. 6, pp. 645–663, 2004.

[134] M. de. Berg, *Computational geometry : algorithms and applications*. Springer, 2008.

[135] C. B. Barber and D. P. Dobkin, “The Quickhull Algorithm for Convex Hulls,” 1996.

[136] M. Adıvar and S. C. Fang, “Convex Analysis and Duality over Discrete Domains,” *J. Oper. Res. Soc. China*, vol. 6, no. 2, pp. 189–247, 2018.

[137] A. C. C. Yao, “A Lower Bound to Finding Convex Hulls,” *J. ACM*, vol. 28, no. 4, pp. 780–787, 1981.

[138] T. M. Chan, “Optimal output-sensitive convex hull algorithms in two and three dimensions,” *Discret. Comput. Geom.*, vol. 16, no. 4, pp. 361–368, 1996.

[139] P. Gedeck, C. Kramer, and P. Ertl, “Computational Analysis of Structure–Activity Relationships,” *Prog. Med. Chem.*, vol. 49, pp. 113–160, Jan. 2010.

[140] I. V. Tetko, “The perspectives of computational chemistry modeling,” *J. Comput. Aided. Mol. Des.*, vol. 26, no. 1, pp. 135–136, Jan. 2012.

[141] S. Weaver and M. P. Gleeson, “The importance of the domain of applicability in QSAR modeling,” *J. Mol. Graph. Model.*, vol. 26, no. 8, pp. 1315–1326, 2008.

[142] N. Fjodorova, M. Novič, A. Roncaglioni, and E. Benfenati, “Evaluating the applicability domain in the case of classification predictive models for carcinogenicity based on the counter propagation artificial neural network,” *J. Comput. Aided. Mol. Des.*, vol. 25, no. 12, pp. 1147–1158, 2011.

[143] F. Sahigara, D. Ballabio, R. Todeschini, and V. Consonni, “Defining a novel k-nearest neighbours approach to assess the applicability domain of a QSAR model for reliable predictions,” *J. Cheminform.*, vol. 5, no. 5, pp. 1–9, 2013.

[144] A. Alexandridis, M. Stogiannos, N. Papaioannou, E. Zois, and H. Sarimveis, “An inverse neural controller based on the applicability domain of RBF network models,” *Sensors (Switzerland)*, vol. 18, no. 1, 2018.

[145] “UCI Machine Learning Repository.” [Online]. Available: https://archive.ics.uci.edu/ml/index.php. [Accessed: 18-Dec-2018].

[146] “Healthcare Dataset Stroke Data | Kaggle.” [Online]. Available: https://www.kaggle.com/asaumya/healthcare-dataset-stroke-data. [Accessed: 10-Aug-2018].

[147] “CRAN - Package rstudioapi.” [Online]. Available: https://cran.rstudio.com/web/packages/rstudioapi/index.html. [Accessed: 31-Jan-2020].

[148] “CRAN Packages By Name.” [Online]. Available: https://cran.r-project.org/web/packages/available_packages_by_name.html. [Accessed: 12-Mar-2020].

[149] I. Hernández-Neuta *et al.*, “Smartphone-based clinical diagnostics: towards democratization of evidence-based health care.,” *J. Intern. Med.*, vol. 285, no. 1, pp. 19–39, Jan. 2019.

[150] I. Sushko *et al.*, “Applicability domains for classification problems: Benchmarking of distance to models for ames mutagenicity set,” *J. Chem. Inf. Model.*, vol. 50, no. 12, pp. 2094–2111, 2010.

[151] P. Brazdil, J. Gama, and B. Henery, “Characterizing the applicability of classification algorithms using meta-level learning,” pp. 83–102, 2012.

[152] I. V Tetko *et al.*, “Critical Assessment of QSAR Models of Environmental Toxicity against,” *Osiris*, pp. 1733–1746, 2008.

[153] E. N. Smirnov, U. Maastricht, and G. I. Nalbantov, “Reliability yields Information Gain,”

Entropy, no. 1, 2002.

[154] M. Kukar and I. Kononenko, "Reliable Classifications with Machine Learning," Springer, Berlin, Heidelberg, 2002, pp. 219–231.

[155] B. Efron and R. Tibshirani, "Improvements on Cross-Validation: The 632+ Bootstrap Method," *J. Am. Stat. Assoc.*, vol. 92, no. 438, pp. 548–560, Jun. 1997.

[156] *,† Arja H. Asikainen, † and Juhani Ruuskanen, and K. A. Tuppurainen‡, "Consensus kNN QSAR: A Versatile Method for Predicting the Estrogenic Activity of Organic Compounds In Silico. A Comparative Study with Five Estrogen Receptors and a Large, Diverse Set of Ligands," 2004.

[157] "SMOTE." [Online]. Available: https://www.cs.cmu.edu/afs/cs/project/jair/pub/volume16/chawla02a-html/node6.html. [Accessed: 24-Jan-2019].

[158] K.-J. Wang, B. Makond, and K.-M. Wang, "An improved survivability prognosis of breast cancer by using sampling and feature selection technique to solve imbalanced patient classification data," *BMC Med. Inform. Decis. Mak.*, vol. 13, no. 1, p. 124, Dec. 2013.

[159] M. Kalinic, "Kernel Density Estimation (KDE) vs . Hot-Spot Analysis - Detecting Criminal Hot Spots in the City of San Francisco," *21st Int. Conf. Geogr. Inf. Sci. (AGILE 2018)*, no. June, pp. 1–5, 2018.

[160] "How to calculate the euclidean distance in R between two matrices each with unequal dimensions - Stack Overflow." [Online]. Available: https://stackoverflow.com/questions/48109002/how-to-calculate-the-euclidean-distance-in- r-between-two-matrices-each-with-uneq. [Accessed: 18-Apr-2019].

[161] N. Gunantara, "A review of multi-objective optimization: Methods and its applications," *Cogent Eng.*, vol. 5, no. 1, pp. 1–16, 2018.

[162] G. Verbeeck, *Optimisation of extremely low energy residential buildings*, no. May. 2007.

[163] M. Czajkowski and M. Kretowski, "A multi-objective evolutionary approach to Pareto-optimal model trees," *Soft Comput.*, vol. 23, no. 5, pp. 1423–1437, 2019.

[164] S. V. Utyuzhnikov, P. Fantini, and M. D. Guenov, "A method for generating a well-distributed Pareto set in nonlinear multiobjective optimization," *J. Comput. Appl. Math.*, vol. 223, no. 2, pp. 820–841, Jan. 2009.

[165] W. Pedrycz, A. Sillitti, and G. Succi, “Computational intelligence: An introduction,” *Studies in Computational Intelligence*, vol. 617. pp. 13–31, 2016.

[166] T. A. Pcr, G. Th, and T. A. P. Express, “Selection of Optimal,” pp. 1–6.

[167] P. B. Brazdil, C. Soares, and J. P. da Costa, “Ranking learning algorithms: Using IBL and meta-learning on accuracy and time results,” *Mach. Learn.*, vol. 50, no. 3, pp. 251–277, 2003.

[168] D. J. (Deborah J. Rumsey, *Intermediate statistics for dummies*. Wiley Pub, 2007.

[169] S. J. Sheather, “Density Estimation,” *Stat. Sci.*, vol. 19, no. 4, pp. 588–597, 2004.

[170] R. R. Bouckaert, “Practical bias variance decomposition,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 5360 LNAI, pp. 247–257, 2008.

[171] E. A. Platanios, A. Blum, and T. Mitchell, “Estimating Accuracy from Unlabeled Data,” *UAI’14 Proc. 30th Conf. Uncertain. Artif. Intell.* , p. 10, 2014.

[172] T. M. Mitchell, “The Need for Biases in Learning Generalizations,” *Readings Mach. Learn.*, no. CBM-TR-117, pp. 184–191, 1980.

[173] S. Geman, E. Bienenstock, and R. Doursat, “Neural Networks and the Bias/Variance Dilemma,” *Neural Comput.*, vol. 4, no. 1, pp. 1–58, Jan. 1992.

[174] Beck, Breindl, and Clark, “QM/NN QSPR models with error estimation: vapor pressure and logP,” *J. Chem. Inf. Comput. Sci.*, vol. 40, no. 4, pp. 1046–51, Jul. 2000.

[175] J. Wiens and E. S. Shenoy, “Machine Learning for Healthcare: On the Verge of a Major Shift in Healthcare Epidemiology,” *Clin. Infect. Dis.*, vol. 66, no. 1, pp. 149–153, 2018.

[176] T. G. Dietterich, “Ensemble Methods in Machine Learning,” pp. 1–15, 2000.

[177] “Statistical and Machine-Learning Data Mining : Techniques for Better Predictive Modeling and Analysis of Big Data, Second Edition.” [Online]. Available: http://web.a.ebscohost.com.brad.idm.oclc.org/ehost/ebookviewer/ebook/bmxlYmtfXzQyNjQ5OF9fQU41?sid=2df6fd28-5f2b-4025-93de-139006669f16@sessionmgr4006&vid=0&format=EB&rid=1. [Accessed: 19-Mar-2019].

[178] O. Khatib, V. Kumar, and D. Rus, *Experimental robotics : the 10th International Symposium on Experimental Robotics*. Springer, 2008.

[179] † Andrew J. Chalk, ‡ and Bernd Beck, and † Timothy Clark*, “A Quantum Mechanical/Neural

Net Model for Boiling Points with Error Estimation," 2001.

[180] U. Kohler and F. Kreuter, *Data analysis using stata*. Stata Press, 2005.

[181] P. Dangeti, *Statistics for machine learning : build supervised, unsupervised, and reinforcement learning models using both Python and R*. .

[182] U. Sahlin, "Uncertainty in QSAR Predictions," *Altern. to Lab. Anim.*, vol. 41, no. 1, pp. 111–125, Mar. 2013.

[183] M. Awad and R. Khanna, "Machine Learning," in *Efficient Learning Machines*, Berkeley, CA: Apress, 2015, pp. 1–18.

[184] S. van Buuren and K. Groothuis-Oudshoorn, "**mice** : Multivariate Imputation by Chained Equations in *R*," *J. Stat. Softw.*, vol. 45, no. 3, 2011.

[185] R. P. Sheridan, "Three useful dimensions for domain applicability in QSAR models using random forest," *J. Chem. Inf. Model.*, vol. 52, no. 3, pp. 814–823, 2012.

[186] C. D. Manning and H. Schütze, *Foundations of statistical natural language processing*. MIT Press, 1999.

[187] S. Banerjee, B. Samynathan, J. Abraham, and A. Chatterjee, "Real-Time Error Detection in Nonlinear Control Systems Using Machine Learning Assisted State-Space Encoding," *IEEE Trans. Dependable Secur. Comput.*, pp. 1–1, 2019.

[188] B. Nithya and V. Ilango, "Predictive Analytics in Health Care Using Machine Learning Tools and Techniques," *Int. Conf. Intell. Comput. Control Syst. ICICCS 2017 Predict.*, pp. 492–499, 2017.

[189] K. Malik, H. Sadawarti, and K. G. S, "Comparative Analysis of Outlier Detection Techniques," *Int. J. Comput. Appl.*, vol. 97, no. 8, pp. 12–21, Jul. 2014.

[190] T.-S. Chou, K. K. Yen, J. Luo, N. Pissinou, and K. Makki, "Correlation-Based Feature Selection for Intrusion Detection Design," in *MILCOM 2007 - IEEE Military Communications Conference*, 2007, pp. 1–7.

[191] H. Ong and V. An, "Mathematical Foundations of Machine," pp. 1–87, 2018.

[192] W. Hu and Y. Tan, "On the robustness of machine learning based malware detection algorithms," *Proc. Int. Jt. Conf. Neural Networks*, vol. 2017-May, pp. 1435–1441, 2017.

[193] "On learning algorithm selection for classification," *Appl. Soft Comput.*, vol. 6, no. 2, pp. 119–138, Jan. 2006.

[194] M. Muaafa, “Multi-Criteria Decision-Making Frameworks for Surveillance and Logistics Applications,” 2015.

[195] R. D.-P. R. Letters and undefined 1996, “A note on comparing classifiers,” *Elsevier*.

Appendix A

Data Analysis

The datasets used in this study are briefly described in this section.

1. Pima Indians Diabetes Dataset

The Pima dataset contains 768 samples from each of two classes of the disease. The dataset includes nine attributes in the dataset: "pregnant", "glucose", "pressure", "triceps", "insulin", "mass", "age", "pedigree", "diabetes".

2. Breast-cancer dataset

The Breast-cancer dataset includes 699 samples from each of two classes of the disease. 11 attributes are contained in the dataset: "Id", "Cl.thickness", "Cell.size", "Cell.shape", "Mitoses", "Marg.adhesion", "Epith.c.size", "Bare.nuclei", "Bl.cromatin", "Normal.nucleoli", "Class".

3. Indian liver patient dataset

The Indian liver patient dataset has 583 samples from each of two classes of the disease. The attributes of the dataset are: "age", "gender", "sgpt", "tot_bilirubin", "direct_bilirubin", "tot_proteins", "albumin", "ag_ratio", "albumin", "alkphos", "Class".

4. Heart dataset

The Heart disease dataset contains 303 samples from each of two classes of the disease. There are 14 attributes in the dataset: "Age", "Sex", "ChestPain", "RestBP", "Chol", "Fbs", "RestECG", "MaxHR", "ExAng", "Oldpeak", "Slope", "Ca", "Thal", "Class".

5. Thyroid dataset

The Thyroid dataset contains 7200 samples from each of three classes of the disease. There are 21 attributes in the dataset: "Age", "Sex", "On_thyroxine", "Query_on_thyroxine", "Sick", "Pregnant" "On_antithyroid_medication", "Thyroid_surgery", "I131_treatment", "Query_hypothyroid", "Query_hyperthyroid", "Lithium", "Goitre", "Tumor", "Hypopituitary", "Psych", "TSH", "T3", "TT4" "T4U", "Class".

6. Cardiotocographic dataset

The Cardiotocographic dataset contains 2130 samples from each of three classes of the disease. There are 25 attributes in the dataset: "b", "e", "LBE", "LB", "AC", "FM", "UC", "ASTV", "MSTV" "ALTV", "MLTV", "DL", "DP", "Width", "Min", "Max", "Nmax", "Nzeros", "Mode", "Mean", "Median" "Variance", "Tendency", "CLASS", "NSP".

7. Hepatitis

The Hepatitis dataset contains 155 samples from each of two classes of the disease. There are 20 attributes in the dataset: "Age", "Sex", "Steroid", "Antivirals", "Fatigue", "Malaise", "Sgot", "Anorexia", "LiverBig", "LiverFirm", "SpleenPalpable", "Spiders", "Ascites", "Varices", "Bilirubin", "Histology", "AlkPhosphate", "AlbuMin", "ProTime", "Class".

Appendix B

1. The robustness of the classifier

In chapter 5, robustness of the classification model based on applicability domain approach is discussed. The model trained using random forests classifier. The experiments on chosen dataset were mainly conducted by using the R language. This work has been conducted using a computer with Windows 10. with Intel® Core™ i7-7th. This work explored several platforms to perform data processing and classification including R, Weka (University of Waikato, 2017). The code of this approach is given below.

```
r=seq(0.1, 0.2, by=0.01);r; i=0

acc<-matrix(c(0),nrow = length (r)+1,ncol = 7,byrow = TRUE)

colnames(acc) <- c("r value","Acc","Ec","Correct class","False
positive","False negative","N.o.Instances")

repeat {

  i=i+1

  D4<-D3[-20] + r[i]

  dim(D4)

  head(D4)

  Newd1<-D4

  dim(Newd1)

  head(Newd1)

  p <- predict(rf, Newd1)

  p1 <- predict(rf, Newd1,type="prob")

  # convex of a new subset

  X <- as.matrix(Newd1)

  chull(X)

  hpts <- chull(X)

  hpts <- c(hpts, hpts[1])

  hpts1 <- chull(data_test[-20])

  hpts1 <- c( hpts1,hpts1[1])
```

```
max(hpts)
min(hpts)
tt<-round(data_test[hpts1,1:19 ],digits = 2)
tt<-cbind(tt,Class=data_test[hpts1,20  ])
tt; cc<-X[hpts, ]
dim(cc); cc
MatrixA1<-cc
#MatrixA2<-as.matrix(X[hpts[1], ])
distancecc = round(rdist(MatrixA1,MatrixA1),digits = 2)
d<-distancecc[1,]
m<-max(d)
ind2 <- which( upper.tri(distancecc,diag=TRUE) , arr.ind = TRUE )
asash2<-data.frame(  val = distancecc[ ind2 ] )
round(asash2,digits = 3)
asash2<-as.matrix(asash2)
ttth2<-cbind(ind2,asash2)
Ne=subset(ttth2,ttth2[,3]==m)
Ne<-Ne[1,]
MatrixA3<-round(as.matrix(data_test[-20]),digits = 2)
distancect = round(rdist(MatrixA1,MatrixA3),digits = 2)
ind2 <- which( upper.tri(distancect,diag=TRUE) , arr.ind = TRUE )
asash2<-data.frame(  val = distancect[ ind2 ] )
round(asash2,digits = 3)
asash2<-as.matrix(asash2)
ttth2<-cbind(ind2,asash2)
ind3 <- which( lower.tri(distancect,diag=FALSE) , arr.ind = TRUE )
asash1<-data.frame(  val = distancect[ ind3 ] )
round(asash1,digits = 3)
asash1<-as.matrix(asash1)
ttth1<-cbind(ind3,asash1)
ttth4<-rbind(ttth1,ttth2)
Ne22=subset(ttth4,ttth4[,1]==1)
Ne33=subset(Ne22,Ne22[,3]<0.95)
Ne55=data_test[Ne33[,2],]
```

```
  Ne44=round(Ne55[-20],digits = 2)
  Ne44=cbind(Ne44,Class=Ne55[,20])
  # Validation
  # Evaluate the model C on D_test #   Compute Accuracy of C
  p <- predict(rf , Ne44)
  p1 <- predict(rf , Ne44,type="prob")
  tab <- table(p, Ne44$Class)
  tab
  acc[i,1]=r[i]
  #correct classification rate
  ac=round( sum(diag(tab))/sum(tab),digits = 2)
  acc[i,2]=ac
  #missclassification rate
  ee=round(1-sum(diag(tab))/sum(tab),digits = 2)
  acc[i,3]=ee
  acc[i,4]=tab[1,1]+tab[2,2]
  acc[i,5]=tab[1,2]
  acc[i,6]=tab[2,1]
  acc[i,7]=nrow(Ne44)
  # on test data
  pp2<-predict(rf,Ne44)
    if (r[i]>0.20){
    break
  }

}
acc
```

2. Obtaining Pareto points

In Chapter 6, a classifier automatically selected using the Pareto set approach of a collection of classifiers obtained from the method of assessing the AD of a classifier (in chapter 4). This is part of the first specific objective. The code of this approach is given below.

```
# for Indian liver dataset
y<-cbind(P.Indian[,2],P.Indian[,4])
```

```
rankIdxList <- fastNonDominatedSorting(y)

rankIdxList


plot(y,col="grey", pch = 19,cex=2,xlab="correctly
Classified",ylab="Threshold",main="Pareto Points in Parameter Space
for thyroid dataset")

l=length(rankIdxList)

s<-rankIdxList[[l]]

s<-sort(s)

points(P.Indian[s,2],P.Indian[s,4],col="red",cex=2, pch = 19)

s<-sort(rankIdxList[[1]])

points(P.Indian[s,2],P.Indian[s,4],type="p",col="red",cex=2, pch =
19)
```

3. Data pre-processing

This is part of all the objectives. The code of data pre-processing is given below.

```
# load libraries

library(mlbench)

library(caret)

# Load data

data(BreastCancer)

# types

sapply(dataset, class)

# summary

summary(dataset)


# class distribution

cbind(freq=table(dataset$Class),
percentage=prop.table(table(dataset$Class))*100)

# convert input values to numeric

for(i in 1:9) {

      dataset[,i] <- as.numeric(as.character(dataset[,i]))

}

# histograms each attribute

par(mfrow=c(3,3))
```

```
for(i in 1:9) {
      hist(dataset[,i], main=names(dataset)[i])
}

# density plot for each attribute
par(mfrow=c(3,3))
complete_cases <- complete.cases(dataset)
for(i in 1:9) {
      plot(density(dataset[complete_cases,i]),
main=names(dataset)[i])
}
# bar plots of each variable by class
par(mfrow=c(3,3))
for(i in 1:9) {
      barplot(table(dataset$Class,dataset[,i]),
main=names(dataset)[i], legend.text=unique(dataset$Class))
}
```