Applying Machine Learning to the Development of an Intelligent Decision Support System for Antimicrobial Prescribing

Done by:
Jamie Kan Yan Ting
B.Eng(*Hons*) Electrical and Electronic Engineering
Imperial College London

A Thesis submitted in fulfilment of requirements for the degree of
Master of Science
in Communications and Signal Processing
of Imperial College London

Department of Electrical and Electronic Engineering
Imperial College London
September 8, 2015

ABSTRACT

Rising antimicrobial resistance is a major problem in the UK and around the world, and one of its main causes is inappropriate prescriptions. This project aimed to achieve effective antimicrobial prescription for critically ill patients, which is especially crucial because antimicrobial therapy must often be administered before laboratory results are available. An Intelligent Decision Support System (IDSS) was developed to model the decision making process of an expert so as to suggest suitable antimicrobial therapy options for a new case. A case is described by its attributes such as age and lactate levels. Using the technique of Case Based Reasoning (CBR), similar cases were retrieved using the K-Nearest Neighbour (KNN) algorithm. A novel approach was used to define the distance metric of the KNN algorithm by taking a weighted average of each attribute's distance. A grid-search algorithm was used to find the optimal set of weights that best represents the relative importance of the attributes. Using the IDSS, the proposed solution had an average specificity accuracy of 94%, showing that the concept of CBR and KNN can be used for antimicrobial prescribing. Also, when the optimal set of weights was used, the sensitivity of the proposed solution increased by 1.7 times, demonstrating the usefulness of weighing the attributes. The accuracy of each antimicrobial drug was found to be directly proportional to how well the drug was clustered when Principal Component Analysis (PCA) was applied. By studying how clustered a drug was and its accuracy of prediction, it was possible to predict antimicrobial drugs that had been overprescribed. Thus, this project contributed to tackling antimicrobial resistance in three ways: firstly, by understanding the expert's decision making process with respect to the relative importance of attributes; secondly, by suggesting suitable antimicrobial therapy options using the IDSS; and thirdly, by examining the drug's characteristics in terms of clustering and accuracy.

ACKNOWLEDGEMENTS

Firstly, I would like to thank my project supervisor Dr Pau Herrero Vinas for his time and guidance throughout the project.

Also, I would like to express my appreciation to Mr Bernard Hernandez for his guidance and support, especially with regards to coding in python.

I would also like to thank my friends Lee Xun Yong, Bryan Huang and Low Guan Jie for their help in brainstorming, validating and improving ideas. Thanks also to Gwendoline Tan for helping me with the medical terminologies and Ng Zhen Yi for her help with LaTeX.

CONTENTS

1

INTRODUCTION

1.1 Project Motivation

Antimicrobial drugs are medicines used against microorganisms such as bacteria, fungi and viruses. Antibiotics are a type of antimicrobial drug that is effective against a single type or multiple types of bacteria. [1]

There is a significant concern regarding antimicrobial resistance, [2] which has been rising over the past few years in Europe. [3] In particular, antibiotic resistance means that certain drugs are no longer effective at inhibiting the growth of the bacteria it was targeted at. Some examples of bacteria that are resistant to antibiotics are Methicillin-resistant Staphylococcus aureus [4], Vancomycin intermediate and resistant Staphylococcus aureus, and Vancomycin-Resistant Enterococci (VRE). [5] Antimicrobial resistance ultimately results in an increase in morbidity and mortality of patients. [6, 7] Misuse of antimicrobial drugs is one of the leading causes of antimicrobial resistance. This can be in the form of unnecessary prescription of antimicrobial drugs, prescribing excessive broad-spectrum drugs or inappropriate prescription of narrow-spectrum drugs, or if critically ill patients are not given timely antimicrobial treatment. [8]

In view of this, this project seeks to achieve prudent usage of drugs by providing individualistic care to each patient. This is exceptionally crucial for critically ill patients because they usually have multiple medical conditions. If these addition concerns are not taken into account, the antibiotic prescribed may be suboptimal for the situation. [9] Furthermore, the cases in the Intensive Care Unit (ICU) are more time-sensitive. Ideally, the most effective drug would be prescribed if the pathogen and its sensitivity has been identified, which is known as ”selective” therapy. However, it usually takes a minimum of 24 hours to identify the pathogen and another 24 hours to obtain its sensitivity. This means that it would take a total of 48 hours to determine which drugs would be the most successful in killing the pathogen. Due to the critical state of the patients in the ICU, the administration of antimicrobials is exceptionally time-sensitive and doctors often have to start antimicrobial therapy before laboratory results are available. This is known as ”calculated” therapy. [10] Unfortunately, 40% of the time, patients are given inappropriate treatment, either due to antibiotics being prescribed unnecessarily when there is no microbial infection, or prescribing a drug that has too broad a spectrum to target the pathogen. [11] This superfluous treatment leads to an increase in antibiotic resistance.

1.2 Aims and Objectives

Aim:

The aim of the project was to develop an Intelligent Decision Support System (IDSS) for antibiotic prescribing. The IDSS would assist doctors in developing individualised prescriptions of antimicrobial drugs for critically ill patients in ICU. This was achieved by presenting the doctor with similar previous cases that are of interest, together with the drug or group of drugs that was prescribed in those cases, as well as how successful their treatment was. It could also recommend a possible antimicrobial therapy based on these retrieved cases. The suggested solution would mimics what an expert would have prescribed.

Objective:

The main objective of the project was to develop an IDSS that, when given a new case, is able to suggest suitable antimicrobial therapy options based on the decision making process of an expert. The IDSS would use the technique of Case Based Reasoning (CBR) together with machine learning to extract similar cases to be presented. In order to create a useful tool for doctors, there must be a good understanding of the medical data used. Specifically, the project seeks to understand the decision making process when an expert prescribes antimicrobial drugs, and to incorporate this into the IDSS. To measure the usefulness of the tool, the accuracy of the prediction was estimated using different sets of parameters, and the best set of parameters was determined.

1.3 Scope

The focus of the project was on critically ill patients who had been warded in the ICU. In particular, the project is targeted at improving the effectiveness of ″calculated″ therapy before laboratory results are obtained.

Also, the desired outcome of the IDSS is to recommend a suitable antimicrobial drug or combination of drugs for a new patient. The aim is to replicate an expert's thought process so as to present alternative antimicrobial treatments that is suitable for a new case received. Because each drug has its own set of dosages, determination of dosage for a drug is not included in the scope of the project. However, this could be considered for an extension of the project in the future.

The Python programming language was chosen as the platform for the system because it is a high-level language that is suitable for Rapid Application Development. The open source Python data analysis library, pandas, was used for data pre-processing. Also, scikit-learn, an open source python machine learning library, was used for data pre-processing and machine learning.

1.4 Report Overview

This report has six main chapters, which are as follows. Chapter 1 covers an introduction to the area of antimicrobial prescription and the problem of antimicrobial resistance, as well as the goals and the focus of this project. Chapter 2 gives the background of Case Based Reasoning (CBR), and also some of the existing approaches used for antimicrobial prescribing. The methodology of the project is described in Chapter 3, which covers the pre-processing of the data, and clustering the data using PCA. Chapter 3 also gives a description of the IDSS developed, and how the performance of the IDSS was measured. In Chapter 4, the results with respect to PCA and the accuracy of the IDSS is discussed. Chapter 5 investigates how the results can be interpreted to help achieve prudent antimicrobial drug prescribing. Finally, Chapter 6 concludes the work done and looks at possible areas of future work.

2

BACKGROUND

2.1 Theoretical Concept: Case-Based Reasoning (CBR)

Case Based Reasoning (CBR) can be defined as applying past knowledge and experiences to solve a new problem. The CBR process can be generalised to a cycle that has 4 stages: Retrieve, Reuse, Revise, and Retain (Figure 1). [12]

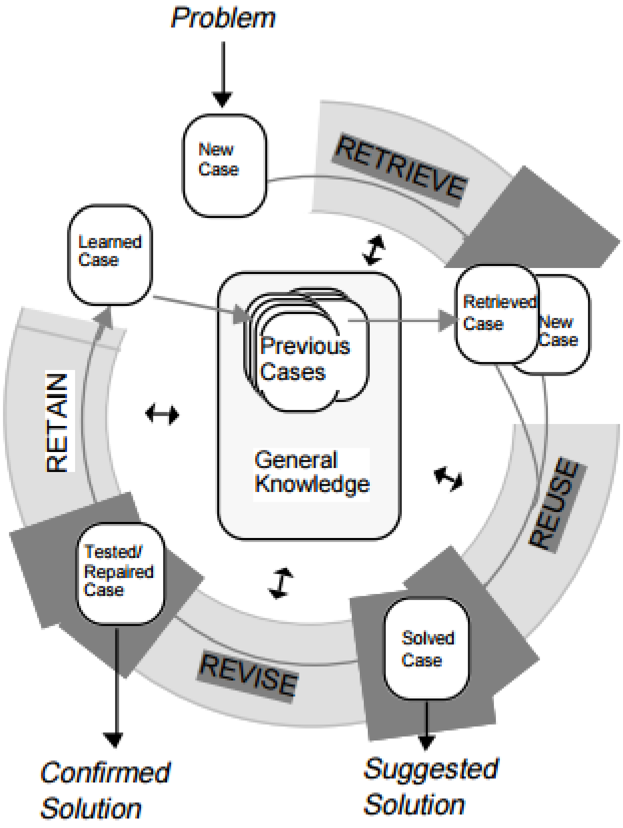


Figure 1: The CBR cycle[12]

When a new problem is received, it is defined as a new case. Then, the following processes occur:

1. Retrieve - From the case base, an identical case is retrieved. If an identical case does not exist, one or more cases that are similar to the new case are retrieved.
2. Reuse - If the retrieved case is identical to the new case, the solution is reused. If the cases are not identical, a solution is formed by adapting the solution from the retrieved case(s). The result of this process is a solved case that is proposed as the suggested solution.
3. Revise - The suggested solution may be tested in a real-world situation. Depending on how successful the suggested solution was, the solved case is revised to give a confirmed solution. Alternatively, the suggested solution may be evaluated by an expert to improve on the solution.
4. Retain - If there has been new knowledge learned from the case, the learned case is added to the case base for future use.

CBR is considered as an alternative to a rule based approach. In a rule based system, there is a knowledge acquisition step which formally represents the knowledge from the previous experiences. However, this may not be easily represented, and requires many previous cases to develop the rules. The advantage of CBR is that it is faster than a rule based approach. This is because it does not require knowledge acquisition. Furthermore, CBR can be implemented even with just a small amount of data initially, and the CBR can be improved as cases are added to the case base over time. [13]

For CBR to be effective in a system, there are a few requirements. Firstly, there must be an underlying model because CBR would not be successful in modelling a random guess. Also, there should be exceptional cases that cannot be easily modelled using a rule based system. Similar cases should also occur frequently, and there is an assumption that similar cases have similar solutions. [13] CBR can be applied in areas that satisfy all the above conditions. Some examples of such areas are law, engineering, and medicine.

2.2 Existing Approaches

This section covers the basic techniques used in antimicrobial prescription, and their limitations. It also explores how computer programs have been successfully used as support systems to doctors. In particular, the ICONS project, from which the idea of this project was developed, is studied in detail.

2.2.1 *Basic Antimicrobial Prescription*

At present, the basic antimicrobial prescription method in most hospitals relies on the expertise of the doctors to determine a suitable therapy. To obtain a solution, two processes are combined. Firstly, there is a rule-based approach, which is objective knowledge derived from textbook knowledge. Secondly, an intuitive approach is used which is based on the doctor's experience. Thus, it is subjective and evolves as the doctor gains experience. [14]

Some hospitals have implemeted programs targetted at reducing antimicrobial resistance. This could be through antibiotic management programs, which are also known as antimicrobial stewardship programs. For example, as part of their Hospital Antibiotic Stewardship program, the Centers for Disease Control and Prevention (CDC) in the U.S. has a checklist to achieve optimal antibiotic prescribing and minimise drug misuse. [15] However, this approach is subjective and time consuming, and is unable to suggest alternatives even if the misuse of a drug has been identified.

2.2.2 *Computerised Decision Support Systems (CDSS)*

The application of Computerised Decision Support Systems (CDSS) to support clinical decisions has made significant advances recently. In [16], CDSS has been defined in a medical context as "any software that directly aids clinical decision making in which characteristics of patients are matched to a computerized knowledge base for the purpose of generating patient-specific assessments or recommendations that are then presented to clinicians for consideration". There are many possible approaches used in CDSS such as rule-based reasoning, Causal Probabilistic Networks (CPN), and case-based reasoning. The success of these methods was evaluated to determine if they were applicable to the context of this project.

Rule-based CDSS

Rule-based CDSS was first introduced in the 1970s, and MYCIN was one of the first such CDSS. In a rule-based system, objective knowledge can be encoded as rules in function-based programs such as: $if \quad a \rightarrow then \quad x$ with a certainty factor $CF(a, x)$. [17] For example, if a patient has a fever, then he might be diagnosed ventilator-associated pneumonia with a confidence of 70%. While this program might be helpful for an inexperienced doctor, it is not ideal because it has a very rigid structure and cannot handle missing information. [18] The paradigm also does not account for the subjective knowledge that experts use to prescribe an optimal therapy. Furthermore, the rules would change over time due to reasons such as the development of antibiotic resistance, and invention of new drugs. However, rules have to be hard-coded and modified manually, making the rule-based paradigm inflexible to the dynamic clinical

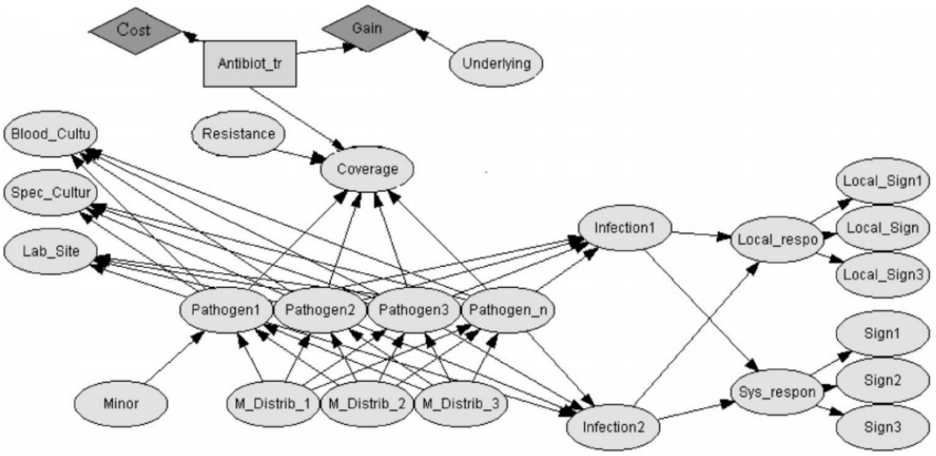


Figure 2: CPN modelling site of infection [21]

environment.

Causal Probabilistic Networks (CPN)

From the concept of certainty factors in MYCIN, the idea of conditional probabilistic networks was developed. [19] The Bayesian network links variables to each other, and the magnitude of each relationship is represented by a probability. One example of how a CPN was used to prescribe antibiotics for moderate and severe infections was the TREAT project. [20] Figure 2 shows the model of the CPN applied in the TREAT project.

For each site of infection, a network model was built. One example of such a site was the urinary track. The network model in Figure 2 showed how **major patient groups** such as urinary catheter(labelled $M_Distrib_1inthefigure)affectedtheprobabilityofaninfe$

CPN was effective in the TREAT project because the project was targeted at ”run-off-the-mill inpatients given antibiotic treatment rather than unusual or rare diseases”. [11, page S94] However, for the context of this project, the conditions of the patients in the ICU are highly complex and diverse, with many combinations and exceptions. It would not be easy to model these combinations and exceptions using the network model, especially if there is insufficient data. Thus, CPN is not appropriate for this project.

CBR in Medical Applications

There has been a number of medical applications of CBR, mainly in the areas of diagnosing diseases, classifying patients, and prescribing or proposing treatments or plans. [22]

One example of a successful implementation of CBR was in diabetics patient management. [23] The objective of the system was to find similar past cases using a nearest neighbour technique. These cases were presented to the doctor to decide on the final treatment prescribed. It is interesting to note that the similar cases could be from the same patient or from another patient. This allows the knowledge pertaining to that patient be retained and passed on even if there was a change in doctors. Furthermore, by using CBR, the knowledge of experts was retained even when they left the hospital. The results showed that a case was assigned to the correct class 83% of the time, and the correct class was in one of the two most likely classes 98% of the time. Thus, this shows that the nearest neighbour technique, together with CBR, is suitable for medical applications and, in particular, finding similar patients.

One of the most significant research done in the area of medical CBR is the ICONS project. [22] Because its context of prescribing antibiotics to patients in ICU is very similar to the scope of this project, the next section will look at the ICONS project in detail.

2.2.3 *ICONS Project*

The ICONS project involved the development of a computer program, ICONS, to reduce antibiotic resistance through antibiotic therapy advice and individualised dosages. In comparison, the objectives of this project and the ICONS project are similar in terms of the context of providing "calculated" therapy for critically ill patients. Thus, the same CBR framework used in ICONS was applied in this project too. However, the retrieval method used in this project was the nearest neighbour algorithm rather than the prototype trees method used in the ICONS project. The reason for the change in retrieval method will be discussed later.

As discussed previously, a doctor uses both objective and subjective knowledge to decide on the most suitable antibiotic therapy for a patient. The CBR technique is well-suited for antibiotic prescribing because it would take into account subjective knowledge. The CBR is also updated easily, thus keeping up with the constant new knowledge being developed clinically. This update takes place whenever a significant number of new cases are added to the system.[14] Thus, the CBR technique was chosen for this project, and more details are provided in Section 3.3.

The ICONS project organised the dataset into a forest of prototype trees in order to retrieve similar cases. Each tree represented a typical antibiotic treatment which corresponded to specific patient characteristics.This was implemented by grouping cases into prototypes according to the ″group of patients″ and ″organ infected″. An example of a prototype tree is ″community-acquired kidney infections″, where the ″group of patients″ is ″community acquired″ and the ″organ infected″ is the kidney. Within the prototype, cases would have different contraindications due to attributes such as allergies to particular drugs, pregnancy, or problems with particular organs such as the kidney and the liver. Since each prototype was expected to be correspond to the same pathological spectrum, and the prescription was then fine-tuned to fit the contraindications. [10]

This method of storing only prototypes reduced the retrieval time and memory storage required even as the number of cases grows over time. However, the disadvantage is that the system is rigid and would not work in some situations. For example, if information regarding the organ infected in a new patient is not available, the system would break down because it would not be able to assign the new case to any prototype. This is in spite of other useful information being available, such as a chest or abdominal examination information, which can be used to locate the problem. Furthermore, each prototype tree and its corresponding pathogen is considered to be independent of each other, which does not accurately reflect how underlying clinical factors may be related. For example, a single pathogen may affect two organs. However, this would modelled as two separate problems as each infected organ corresponded to a different, independent prototype tree. In such a situation, the ICONS project would not be able to consider a holistic solution to the problem which was caused by a single underlying pathogen.

In the ICONS project, the nearest neighbour algorithm was disregarded as a retrieval technique. This was because the nearest neighbour algorithm is only appropriate for metric values, but several of the attributes in the case description are categorical data such as allergies and pregnancy. Although these categories could be numbered, they were unordered, and thus the nearest neighbour algorithm was deemed inappropriate. [10] In this project, the problem of unordered numerical data was overcome by using the similarity functions, and in particular the Equals function described in Section 3.4.1.

3

METHODOLOGY

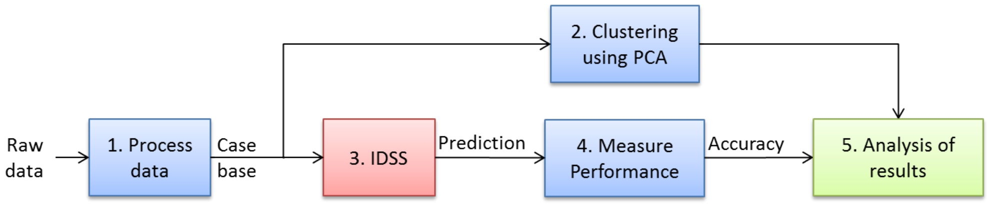


Figure 3: Overview of methdlology

This chapter covers the four main processes in the methodology of this project. Firstly, the raw data is processed to form the case base (Section 3.1). Secondly, Principal Component Analysis (PCA) is applied to the processed data to better understand the data in order to access if the case base is suitable for CBR (Section 3.2). Thirdly, the development of the IDSS is described in detail in Section 3.3. The IDSS uses the processed data as its case base. Within the IDSS, two techniques were described in detail: the retrieval technique using K-Nearest Neighbours (KNN) in Section 3.4, and, in Section 3.5, the classification methods used to propose a solution. Finally, the accuracy of the suggested solution is measured in terms of how well it is able to mimic what an expert would have prescribed (Section 3.6). The performance of the system, together with the results of the PCA analysis, would be analysed in the next chapter.

3.1 Data Processing

The medical data used was collected mainly from the ICU of Charing Cross Hospital and Imperial College Healthcare NHS Trust between August 2014 and April 2015. There was a total of 733 cases collected from 282 patients. Each case consists of a case description containing different attributes(e.g. patient age, body temperature, ventilation support), a solution which is the antimicrobial therapy prescribed by the doctor, and the outcome of the implemented solution. Although there are 37 attributes used to describe the case, some attributes could not be considered because there was insufficient data collected for that attribute. A list of the parameters and the amount of data available for each parameter is given in Appendix A. In the future, if more data has been collected for these attributes, these parameters can be added into the system to give a more complete description of each case and improve the prediction accuracy overall.

Nine attributes were used in the case description. There were two main types of attributes, namely numerical data and categorical data. (Table 1). For numerical data, the attributes considered are 'Patient Age', 'Oxygen Requirements', 'Respiratory Rate', and 'Lactate'. For categorical data, the attributes considered are 'Abdominal Examination', 'Ventilation Support', 'Chest Radiography', 'Chest Examination', and 'Urinary Catheter'. The categorical data was encoded as described in Section 3.1.1 to ensure compatibility with any subsequent functions used. Also, the data is filtered with respect to the drug or combination of drugs prescribed.

No.	Attribute	Type	Range
1	Patient Age	Numerical	$0 - 115$
2	Oxygen Requirements	Numerical	$0.21 - 0.95$
3	Respiratory Rate	Numerical	$6 - 56$
4	Lactate	Numerical	$0.1 - 23$
5	Abdominal Examination	Categorical	SNT (soft non-tender), Tender
6	Ventilation Support	Categorical	Extubated, Intubated, NIV, Own, Tracheo
7	Chest Radiography	Categorical	Air under diaphragm, Clear, Congestion, Consolidation, Effusion, None
8	Chest Examination	Categorical	Clear, Crackles, Dull, None, Wheeze
9	Urinary Catheter	Categorical	Yes, No

Table 1: Attributes in the case description

3.1.1 *Encoding*

If each category in the attribute is mutually exclusive, the attribute can be encoded using either a label encoder or one-hot encoding, and this is summarised in Table 2.

The label encoding method was implemented in python using sklearn.preprocessing.LabelEncoder. For an attribute which has C classes, the classes would be labeled from 0 to $C - 1$, in alphabetical order. For example, for the attribute 'Chest Examination', there are 5 classes, which are labeled 0 to 4. This method of encoding is suitable when using the Equals function described in Section 3.4.1 because the Equals function does not assume any order to the labels. However, for other functions, this method cannot be used as it assumes that the class labeled 0 is closer to class 1 than to class 5, which may not be true. In this case, one-hot encoding is required.

Attribute(s)	Label Encoding	One-Hot Encoding
Renal Support: {'CVVH', 'None'}		
'CVVH'	0	[1, 0]
'None'	1	[0, 1]
Chest Examination: {'Clear', 'Crackles', 'Dull', 'None', 'Wheeze'}		
'Clear'	0	[1, 0, 0, 0, 0]
'Crackles'	1	[0, 1, 0, 0, 0]
'Dull'	2	[0, 0, 1, 0, 0]
'None'	3	[0, 0, 0, 1, 0]
'Wheeze'	4	[0, 0, 0, 0, 1]
Example: [Age, Renal Support, Chest Examination]		
Patient 1:		
[30, 'None', 'None']	[30, 1, 3]	[30, 0, 1, 0, 0, 0, 1, 0]
Patient 2:		
[70, 'CVVH', 'Crackles']	[70, 0, 1]	[70, 1, 0, 0, 1, 0, 0, 0]

Table 2: Examples of Age, Renal Support and Chest Examination encoded using Label Encoding and One-Hot Encoding

For the case of drugs prescribed, a multilabel binarizer is used because a patient can be prescribed more than one drug. For example, there may be a total of 4 possible drugs in the entire case base: Meropenem, Metronidazole, Tazocin, and Vancomycin. A binary matrix is used to represent the drugs prescribed; if a drug is prescribed, its corresponding element in the array is set to 1, if not, the element is 0. Some possible prescription combinations are shown in Table 3. This method was similar to one-hot encoding, but for multiple labels. It was implemented using sklearn.preprocessing.MultiLabelBinarizer. Multilabel classification must be used with multilabel data. This concept is explained in detail in Section 3.5.3.

This multilabel technique should also be used for labelling drugs that a patient is allergic to. Due to very few cases having information regarding drug allergy, this attribute was not used in the current program. However, this should be noted for future developments of the program.

Drugs: {'Meropenem', 'Metronidazole', 'Tazocin', 'Vancomycin'}	
Prescription	multilabel Binarizer
['Tazocin']	$[0, 0, 1, 0]$
['Meropenem']	$[1, 0, 0, 0]$
['Meropenem', 'Vancomycin']	$[1, 0, 0, 1]$
['Tazocin', 'Metronidazole']	$[0, 1, 1, 0]$

Table 3: Examples using multilabel binarizer for a set of 4 drugs

3.1.2 *Filtering*

The data had to be filtered before it was suitable for use in the IDSS. In particular, rarely prescribed drugs were removed from the data set because there was insufficient data to accurately conclude the drug's usage. Although there were 36 unique drugs prescribed in total in the data, Moxifloxacin, Ertapenem, Ethambutol and Cefalexin occurred only once, and Teicoplanin and Benzylpenicillin only occurred twice in the entire data set. There are 14 drugs that occured less than 10 times, and 8 drugs that occurred between 10 and 30 times in the data. These drugs were removed by setting a lower threshold T_L.. If $P_x < T_L$, where P_x was the number of times drug x is prescribed, then drug x was removed from the case base. Then, if a case had no drug prescribed, that case was removed from the data because it did not belong to any class.

Case No.	Drugs Prescribed Before Filtering	Drugs Prescribed After Filtering
1	Tazocin	Tazocin
2	Tazocin	Tazocin
3	Tazocin	Tazocin
4	Meropenem	Meropenem
5	Meropenem	Meropenem
6	Cefalexin, Meropenem	Meropenem
7	Tazocin, Meropenem	Tazocin, Meropenem
8	Ethambutol	[deleted]

Table 4: Filtering cases so that infrequently prescribed drugs are removed

In Table 4, there is a toy example where Tazocin and Meropenem are frequently prescribed drugs, while Cefalexin and Ethambutol are infrequently prescribed drugs. Cefalexin and Ethambutol are removed from the data set. As a result, only Meropenem was prescribed for case 6, and case 8 was completely deleted. Note that in case 7, although the combination of prescribing both Tazocin and Meropenem only occurs once, since both drugs are frequently occurring, the case is left unchanged.

Filtering is also used to remove drugs that occur too frequently in the data as these drugs might bias the data. This hypothesis was tested in Section 4.2.1. This is done by setting an upper threshold, T_H, and if $P_y > T_H$, drug y is removed from the case base.

Additional filtering can be performed with regards to the number of drugs prescribed per patient. This is useful because if only cases with one drug prescribed are chosen, both multiclass and multilabel classification can be applied. With respect to the earlier example in Table 4, after such a filtering, cases 6 and 7 would be removed because both have more than one drug prescribed.

3.2 Data analysis using Principal Component Analysis (PCA)

From Section 2.1, in order for CBR to be effective in a clinical setting, there must be an underlying relationship between the case description and antimicrobial therapy described. However, this relationship cannot be easily inferred from the 9 attributes in the case description. Thus, Principal Component Analysis (PCA) is used to reduce the high-dimensional data to fewer dimensions. This is done by projecting the data onto a lower dimensional space in a way that maximises variance of the projected data. [24] By removing redundant dimensions, a highly complex problem is simplified, and hidden structures in the data may be revealed. [25]

Mathematical derivation of PCA

For the IDSS, there were N cases, which were described by $n = 9$ attributes that had been encoded. This gives an $N \times n$ data matrix, D.

$$D = \begin{bmatrix} d_{11} & d_{12} & \dots & d_{1n} \\ d_{21} & d_{22} & \dots & d_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ d_{N1} & d_{N2} & \dots & d_{Nn} \end{bmatrix}$$

For each attribute (represented by the columns in D),the values are normalised so that none of the attributes were over-represented. The resultant matrix is U. Then, Σ, the covariance matrix of U, is found using

$$\Sigma = \frac{U^T U}{N-1}$$

The eigenvalues λ and their corresponding eigenvectors x of Σ are found using Eigendecomposition. In matrix form, this can be expressed as:

$$\Sigma\Phi = \Phi\Lambda$$

$$\Sigma = \Phi \Lambda \Phi^T$$

where

$$\Phi = [x_1, ..., x_n], \ \Lambda = \begin{bmatrix} \lambda_1 & & & \mathbf{O} \\ & \lambda_2 & & \\ & & \ddots & \\ \mathbf{O} & & & \lambda_n \end{bmatrix}$$

If Λ is normalised, the eigenvectors form a set of orthonormal basis vectors. For PCA, the eigenvectors are sorted from largest to smallest, and the first m vectors are used to reduce dimensionality to m.

$$\Sigma_{new} = [x_1, ..., x_m], \ m \leq n$$

The eigenvectors chosen form the principal components, and these principal components are the axes on which the data is presented. [26]

PCA was implemented in python using sklearn.decomposition.PCA. From the results described in Section 4.1, it was concluded that the case base can be clustered, implying the necessary relationship between clusters and drugs prescribed. This provided the basis and justification to apply CBR to the clinical dataset used in this project.

Note that PCA was not used in the IDSS because it is a lossy compression. Furthermore, the axes in PCA do not have any physical or clinical meaning. Thus, if PCA was used, the set of weights allocated to each axis does not give any additional insights into the decision making process of an expert. In contrast, by using the attributes as axes, an attribute assigned with a higher weight would imply that the attribute was more important to the decision-making process. Hence, PCA was only used in the project as a tool to better understand the data in the case base.

3.3 Overview of IDSS

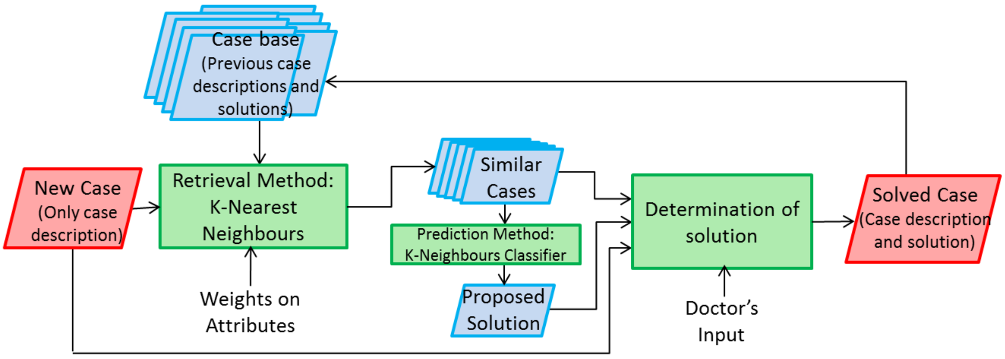


Figure 4: Overview of IDSS

The IDSS is developed using the CBR technique and shown in Figure 4. The process is as follows:

1. A new problem is received as a case description that contains the attributes of the patient. The objective is to find a suitable solution, in this case an antimicrobial treatment, for the new case.

2. Similar cases from the case base are retrieved using the K-Nearest Neighbours (KNN) algorithm, as described in Section 3.4. In this step, a novel approach was applied to measure the similarity between cases. Each attribute of the new case was compared to a previous case. The difference of the attribute between the new and previous cases was quantified and normalised between 0 and 1. A weighted sum of the differences between all the attributes was calculated to give the overall distance between the new case and its neighbouring, previous cases. Thus, the K-Nearest Neighbours were the K most similar cases from the case base which had the lowest overall distance to the new case.

3. Given the *K* most similar cases, a case solution is proposed. The solution is the most likely antibiotic therapy that would have been prescribed by an expert. This is the "Reuse" process in the CBR cycle in Figure 1 and it is described in Section 3.5.

4. The proposed solution is presented to the doctor together with the most similar cases so that the doctor can understand the reasoning behind the suggestion. He can then compare the case descriptions of the similar cases with that of the current case and decide on the actual antibiotic therapy to be implemented.

5. The implemented solution and its outcome (how successful the antibiotic therapy was) is added to the new case's description to form a solved case. The solved case is added into the case base.

3.4 Retrieval Technique: K-Nearest Neighbours (KNN)

The k-Nearest Neignbours (KNN) algorithm is a commonly used retrieval method for CBR and was chosen was for the IDSS as it has been successfully used for previous medical CBR research. [23] It was implemented in python using sklearn.neighbors.NearestNeighbors.

For most KNN algorithms, the metric used is Euclidean distance. The smaller the Euclidean distance, the "nearer" the "neighbour", implying a higher similarity. In the IDSS developed in this project, an alternative metric for distance was applied to cater specifically to the nature of the attributes in the case description. Although this metric is based on an existing framework, to the author's best knowledge, the application of the metric to assist in antimicrobial prescribing was novel.

3.4.1 *Novel Application of Distance Metric*

Similarity functions were developed based on the KNN similarity framework from jCOLIBRI, which is a reference platform for CBR using java. [27] The similarity functions Equals and Interval were used for categorical and numerical attributes respectively. The output of these functions were then combined using a weighted average.

Equals Function

The Equals function is used for comparing categorical attributes like chest examination. It is based on the Equals local similarity function in jCOLIBRI. The output of the function is 1 if the attributes in the new case is the same as the retrieved case, and 0 otherwise.

When the Equals function is used, there is an assumption that each of the categories are dissimilar. This is true for some attributes with only two possible categories such as urinary catheter with only 'yes/no' categories. However, this may not true for other attributes. For example, if a new case's 'Chest Examination' is "Clear" which means that there is no problem detected, this is similar but not exactly the same as if the retrieved case belongs to the category "None", which is when there is no chest examination done because there is no suspected problems in that area. Using the Equals function, the output is 0, and this similarity is not captured. However, because it is difficult to quantify the similarity between the categories, the Equals function is used as a basic comparison.

Interval Function

The Interval function is used for numerical attributes. It is based on the local similarity function of the same name in jCOLIBRI. For each attribute such as age, the difference between the values in the new and previous cases were calculated and normalised by the range of that attribute.

$$Interval_i = 1 - \frac{|NewCaseValue - PreviousCaseValue|}{Range_i} \tag{1}$$

The Interval function normalises the distance so that the output lies between 0 and 1. The higher the output, the more similar the new case is to the retrieved case.

Combining Attributes

In the IDSS, there are nine different attributes that are combined together to give a similarity score. After comparing each of the attributes using the Equals or Interval function, the output similarity of each attribute, x_i is combined using a weighted average equation as shown below. This concept is based on the global similarity function in jCOLIBRI. All the weights w of the attributes lie between 0 and 1, and the weights sum to 1. Thus, the similarity score was also in the range of 0 to 1 with a higher score representing a higher similarity. If a new case is identical to a retrieved case, the similarity score will be 1. This is summarised in Equation 2.

$$Similarity = \sum_i w_i x_i, \ where \ i = [1, 2, ...N], \ \sum_i w_i = 1, \ 0 \leq x_i \leq 1 \qquad (2)$$

In the KNN algorithm, the nearest neighbours are found by choosing those with the smallest distance. The distance is calculated using Equation 3. The more similar a retrieved case is to the new case, the smaller the distance, and the more likely the retrieved case is one of the k-nearest neighbours to the new case. If the retrieved case and new case is identical, the distance score will be 0.

$$Distance = \sum_i w_i (1 - x_i) = 1 - Similarity \qquad (3)$$

Case No.	Age (range: 0 to 115); weight=0.2	Ventilation Support weight=0.8	Drug Actually Used	Similarity score (Eq. 2)	Distance score (Eq. 3)
New	60	Intubated	?	-	-
1	55	Own	A	$0.2 \times (1 - \frac{\|60-55\|}{116}) + 0.8 \times 0 = 0.19$	0.81
2	70	Intubated	B	$0.2 \times (1 - \frac{\|60-70\|}{116}) + 0.8 \times 1 = 0.98$	0.02
3	20	Intubated	B	$0.2 \times (1 - \frac{\|60-20\|}{116}) + 0.8 \times 1 = 0.93$	0.07

Table 5: Attributes in the case description

A simplified example of how a new case is compared to 3 retrieved cases is given in Table 5 where the two attributes of Patient Age and Ventilation Support are considered. It can be seen that although the 1^{st} retrieved case is the closest to the new case in terms of age, it is the furthest from the new case overall. This is because it has a different ventilation support, and this attribute has a much higher weight than age. Both Cases 2 and 3 are a very small distance away from the new case because they both have the same ventilation support

(Intubated) as the new case. Case 2 is nearer to the the new case than Case 3 because the age of the patient in case 2 is closer to that of the new case.

These weights give an indication of the significance of the attribute. The higher the weights, the more important the attribute. This can be seen clearly in the simple example in Table 5 where 'Ventilation Support' is considered more important than 'Patient Age'. Thus, cases with the same 'Ventilation Support' categories are considered more similar, and 'Patient Age' is used as a refinement to distinguish between cases with the same 'Ventilation Support'.

3.4.2 *Finding Optimal Weights*

The "best" set of weights was the set of weights that best represented the "similarity" of cases. Cases that were prescribed the same drugs were defined to be similar. This was a reasonable assumption given the results of the PCA in Section 4.1. For the example in Table 5, if the expert's opinion is that the new case should be prescribed drug B, then the weights chosen are suitable. However, if the new case was supposed to be prescribed drug A, then the set of weights chosen above was incorrect, and the 'Age' attribute should have a higher weight than 'Ventilation Support'.

A grid search algorithm was used to find the "best" set of weights. Quantitatively, this would be the set of weights that maximised the accuracy (measured using the equations in Section 3.6.2). The disadvantage of the grid search algorithm is its high complexity, which translates into a long running time required to find the ideal weights.

One alternative method considered was multiple linear regression, which finds the relationship between n (in this case, $n = 9$) attributes and a response variable Y, which would be the similarity between the set of drugs prescribed. [28]. This is expressed in Equation 4, where Y is the similarity between drugs and w_i is the weights corresponding to the i^{th} attribute, represented as x_i. The technique would find the values of w_i as well as the arbitrary constant, c, that best satisfy Equation 4.

$$Y = \sum_i w_i x_i + c \tag{4}$$

At first glance, this equation seems similar Equation 2. However, this method is not compatible with the concept of KNN and distances. This is because in multiple linear regression, the weights can take on any value, including negative values. A negative weight would mean that the more different the cases are with respect to an attribute, the more likely they would be prescribed the same drug. This does not reflect the decision making process of an expert in a clinical setting. Also, there is no physical interpretation of the arbitrary constant. Thus, although multiple linear regression may be able to model the data statistically, this does not satisfy this project's objective of modelling and interpreting the decision making process of the expert.

Although the grid search algorithm is computationally intensive, this is not a significant limitation to the actual implementation of the IDSS. This is because once the "best" set of weights is found, only this one set of weights is used each time a new case comes in. The entire grid search process would only need to be repeated occasionally to update the best set of weights when the case base changes significantly.

3.5 Proposing a Solution

After retrieving the K most similar cases, a proposed solution is formed by adapting the solutions from the retrieved cases. This adaptation is done using the process of K-Neighbours Classifier. This can be done using either multiclass or multilabel classification.

3.5.1 *K-Neignbours Classifier (KNC)*

KNN can be used to as a classifier to adapt a solution for the new case. The K-Neighbours Classifier (KNC) was implemented using sklearn.neighbors.KNeighborsClassifier. The concept of KNC is a simple majority voting: if at least half of the nearest neighbours belong to class C, the case is assigned to class C. As an illustration, a toy example of using KNN to predict the class of an unknown case is shown in Figure 5.

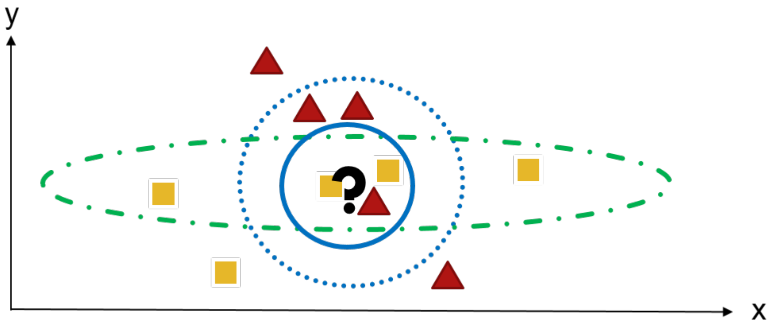


Figure 5: Using KNN to predict the class of a new case, "?"

In Figure 5, there are 2 dimensions, representing 2 attributes, x and y. The distance to the query case is measured using the Equations 2. The blue circles represent the situation where the x and y dimensions have equal weights of

0.5 each. If $k = 3$ (inside the solid blue circle), the case is deemed to be of the "yellow square" class using majority voting. However, if K is chosen to be 5 (inside the dotted blue circle), then the case belongs to the "red triangle" class. Thus, there is a need to choose the most suitable k for the model. This was a hyperparameter optimization problem that was solved using a grid search algorithm.

Another factor that affects the prediction is the value of the weights for each attribute, as described in Section 3.4.1. In the toy example, the green oval represents the situation where the attribute y is assigned a higher weight than x. Then if $k = 5$, 4 out of the 5 nearest neighbours are of the "yellow square" class. the new case is now classified to be "yellow square". This is different from when equal weights are used, and the number of neighbours is also 5.

3.5.2 *Multiclass Classifier*

In the IDSS, there are at least 20 different drugs that occur more than 10 times in the case base. If the system only considers the case where only one drug is prescribed per case, the multiclass classifier can be use. Each class represents one drug, and there are more than 2 classes. A one-versus.-rest classifier (also known as one-versus.-all) is used for multiclass classification. The one-vs.-rest classifier is shown in Figure 6.

A single classifier is trained per class. First, class A is treated as the positive class and the remaining classes are negative classes. This is repeated with class B as the positive class and the rest as the negative class, and so on. [29]

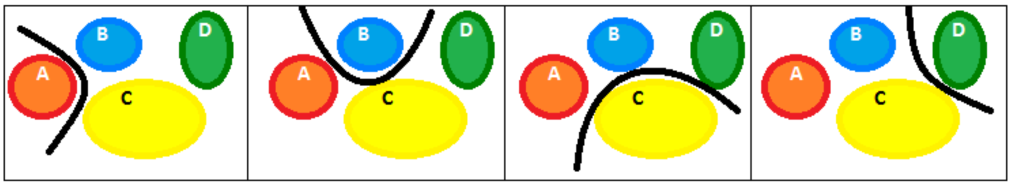


Figure 6: Using one-vs.-rest method to classify four classes

In the previous example in Figure 5, there were only two classes. Thus, if k is odd, at least half of the nearest neighbours would belong to one class. However, this is not true for a multiclass classifier. There are situations where a prediction cannot be made when using a multiclass classifier. For example, if $k = 7$, out of the 7 neighbours a case has, 3 of the neighbours may have been prescribed drug A, while the other 4 cases are prescribed drugs W, X, Y, and Z. Although drug A is prescribed the most number of times compared to the other drugs, a prediction is not made because it was not prescribed more than half the time.

Thus, the IDSS was tested using two possible configurations for the KNC:

1. **No forced prediction**: A prediction is only made if at least half the neighbours are of a certain class. This is the default configuration, and a prediction would not have been made for the above example.

2. **With forced prediction**: A case is allocated the class or classes which have the most number of votes, even if less than half its neighbours belong to that class. For the example above, drug *A* would be predicted.

3.5.3 *Multilabel Classifier*

The multiclass classification is only suitable when one drug is prescribed per case. However, there are many cases where more than one drug was prescribed to a single case. Out of the 733 cases in the data used, there were 36 unique drugs, but 238 possible combinations. It is not possible to give each combination a class as there would be insufficient data per combination, and many combinations are very rare. As such, multilabel classification was favoured over multiclass classification when more than one drug is prescribed per case.

After encoding the data using the multilabel binarizer as described in Section 3.1.1, a single classifier is trained per label. Using the example in Table 3, the classifier is first trained for Meropenem (either prescribed Meropenem or not), then for Metronidazole, and so on. In the case of multiclass classification, if a prediction is made, the suggested solution for a case would be exactly one drug. However, if the multilabel classifier is used, a proposed solution could contain more than one drug, or no drugs at all. For the case when only one drug was prescribed per case, both the multiclass and multilabel classifier were tested to find the most appropriate classifier for the system.

3.6 Measuring the Performance of the IDSS

In order to measure the performance of the IDSS and determine the best set of parameters and configurations, cross validation is used. This allows testing to be done using the existing case base.

Figure 7 gives an overview of how the performance of the IDSS is measured. Some cases in the case base are chosen to form a testing set. Instead of a new case, the cases in the testing set are treated as ”new cases”. The rest of the case base forms the training set. By using only the case descriptions of the testing set, a solution is suggested. This proposed solution is compared to the actual solution which was the drugs prescribed by the experts for those cases.

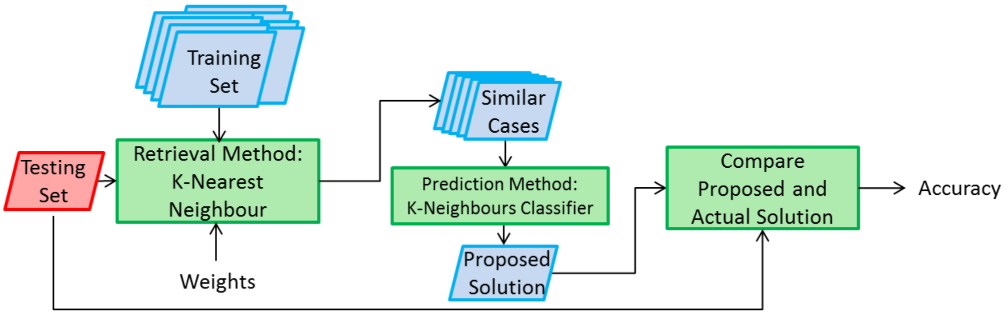

Training Set
Testing Set
Retrieval Method: K-Nearest Neighbour
Weights
Similar Cases
Prediction Method: K-Neighbours Classifier
Proposed Solution
Compare Proposed and Actual Solution
Accuracy

Figure 7: Measuring IDSS performance

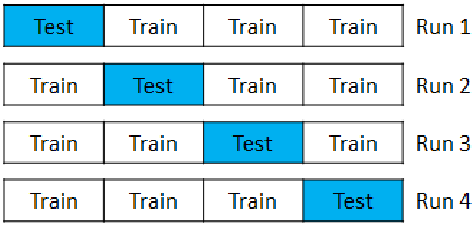
Two main measures were used as performance metrics: the accuracy per case, and the accuracy per drug. Different configurations (e.g. different number of neighbours, k; multiclass vs. multilabel classification) were tested using these metrics to determine the best configuration for the IDSS.

3.6.1 *Stratified K-folds Cross Validation*

Cross validation, also known as rotation estimation, involves splitting the data into a testing and a training set, and performing tests multiple times using different partitions. The output accuracy is averaged over all the runs.

K-folds

Because of the large number of cases, the K-fold technique was used for cross validation. This means that the case base is split into K groups of equal sizes (known as folds). On the first run, the first fold is used as the testing set while the rest is used as the training set. This is repeated a total of K times, and each time a different fold is used as the testing set. Figure 8 shows the K-fold cross validation process for 4 folds. Note that the K here should not be confused with the K in the KNN algorithm; these Ks are unrelated.



Test	Train	Train	Train	Run 1
Train	Test	Train	Train	Run 2
Train	Train	Test	Train	Run 3
Train	Train	Train	Test	Run 4

Figure 8: K-folds cross validation for 4 folds

Stratified K-folds

In the standard K-folds, the cases are usually assigned to the folds randomly. A variation of K-folds is the stratified K-folds, which ensures that the percentage of each class or label represented in each fold remains approximately the same. This was the technique used for the IDSS. This is because some drugs are only prescribed very rarely. If all of the cases that were prescribed drug x were in the testing set, it would be impossible that any of the neighbours assigned to the testing case would be of class x. Thus, stratified K-folds was used to prevent such a situation.

Choice of K for K-folds

It is desirable for K to be large as the larger K is, the more closely the training set resembles the actual case base. However, K cannot be too large in this application because some classes of drugs are very rare. For every class to be represented at least once in each fold, K cannot be too large. Furthermore, the larger$K$is, the longer the running time and the higher the variance between the results of each run. Given these considerations, the number of folds, K, was chosen to be 7 for the testing of the IDSS.

3.6.2 *Calculation of Accuracy*

The accuracy of the proposed solutions was measured using two performance metrics: the accuracy per case, and the accuracy per drug. A higher accuracy means that the IDSS is better able to model the decision making process of the expert.

Accuracy per Case

For each testing case, the $\{Hypothesis\}$ is defined as the set of drugs that is proposed by the IDSS, and the $\{Solution\}$ is the set of drugs that was actually prescribed by the expert for thae testing case. The accuracy per case is defined as:

$$Accuracy = \frac{Cardinality\ of \Big\{ \{Hypothesis\} \cap \{Solution\} \Big\}}{Cardinality\ of \Big\{ \{Hypothesis\} \cup \{Solution\} \Big\}} \tag{5}$$

Some examples of proposed solutions (Hypothesis) and drugs prescribed (Solution) are given in Table 6.

From Table 6, it can be seen that the accuracy is penalised when the hypothesis does not contain a drug that was actually in the solution (No. 2). In order to prevent the suggest solution simply containing as many drugs as possible, there is also a penalty for suggesting a drug that was not prescribed by the expert (No. 6).

No.	{Hypothesis}	{Solution}	$\left\{\{Hypothesis\} \cap \{Solution\}\right\}$	$\left\{\{Hypothesis\} \cup \{Solution\}\right\}$	Accuracy (Eq. 5)
1.	{T, M}	{T, M}	{T, M}	{T, M}	2/2=100%
2.	{T}	{T, M}	{T}	{T, M}	1/2=50%
3.	{A, M}	{T, M}	{M}	{A, T, M}	1/3=33.3%
4.	{T}	{M}	{}	{T, M}	0/2=0%
5.	{T, M}	{A, C, T}	{T}	{A, C, T, M}	1/4=25%
6.	{A, C, T, M}	{T}	{T}	{A, C, T, M}	1/4=25%
7.	{}	{T, M}	{}	{T, M}	-

Table 6: Calculation of accuracy per case

Finally, the overall accuracy is the mean of the accuracy per case, over all the cases. Note that when there is no prediction (possibly due to the multiclass classification in Section 3.5.2), that case is not counted to the overall accuracy. This is not the same as when the hypothesis is completely wrong and the accuracy is 0% (No. 4). The overall accuracy for the all the cases in Table 6 is $\frac{100+50+33.3+0+25+25}{6} = 38.8\%$. Note that No. 7 is not counted because no prediction was made.

Accuracy per Drug

By using multilabel classification, it is easy to calculate the accuracy per drug because each drug is a binary classification. Thus, the measures of prediction for binary classification have been adapted as performance metrics to measure how accurate the prediction of each drug is. Three measures are chosen to give the most insight into the data: the Sensitivity, Specificity and Positive Likelihood ratio.

For the application of the IDSS:

- True Positive (TP): Prediction = 1 and Solution = 1.
  Drug is suggested by the IDSS and was prescribed by an expert.
- False Positive (FP): Prediction = 1 and Solution = 0.
  Drug is suggested by the IDSS but was not prescribed by an expert.
- True Negative (TN): Prediction = 0 and Solution = 0.
  Drug is not suggested by the IDSS and also not prescribed by an expert.
- False Negative (FN): Prediction = 0 and Solution = 1.
  Drug is not suggested by the IDSS but was prescribed by an expert.

Sensitivity measures how likely that if an expert prescribes drug A, the IDSS also recommends drug A as a solution. It is measured by the True Positive Rate (TPR), as shown in Equation 6.

$$Sensitivity = TPR = \frac{No.\ of\ TP}{No.\ of\ TP + FN} \tag{6}$$

Specificity measures how likely a drug is not recommended by the IDSS, given that it was not prescribed by the expert. It is measured by the True Negative Rate (TNR), as shown in Equation 7.

$$Sensitivity = TNR = \frac{No.\ of\ TN}{No.\ of\ TN + FP} \tag{7}$$

Ideally, every drug should have both a high sensitivity and a high specificity. However, there is a trade-off between sensitivity and specificity. In order to quantify this trade-off to find the optimal point, the aim is to maximise the Positive Likelihood Ratio (LR+). Equation 8 shows the calculation of LR+.

$$LR+ \ = \frac{Sensitivity}{1 - Specitivity} = \frac{TPR}{1 - TNR} \tag{8}$$

4

RESULTS

In this chapter, the results from the PCA and the IDSS are detailed, and the relationship between them is analysed. Also, the different configurations of the IDSS are investigated. In particular, it was important to find a suitable number of neighbours, K, and the optimal weights to be allocated.

4.1 Clustering using PCA

The purpose of PCA was to reduce the dimensionality of the data so as to find out if the cases could be clustered based on their case descriptions, and if there was a relationship between the clusters and the antimicrobial drug prescribed. The attributes used in the case description were the same nine that are described in Table 1. From Section 3.2, each attribute, represented by the columns of the data matrix, must be normalised. Two normalisation methods were tested: the standard scaler normalised the mean to 0 and variance to 1, and the Minmax scaler normalised all the values to lie in the range of 0 to 1.

Standard Scalar

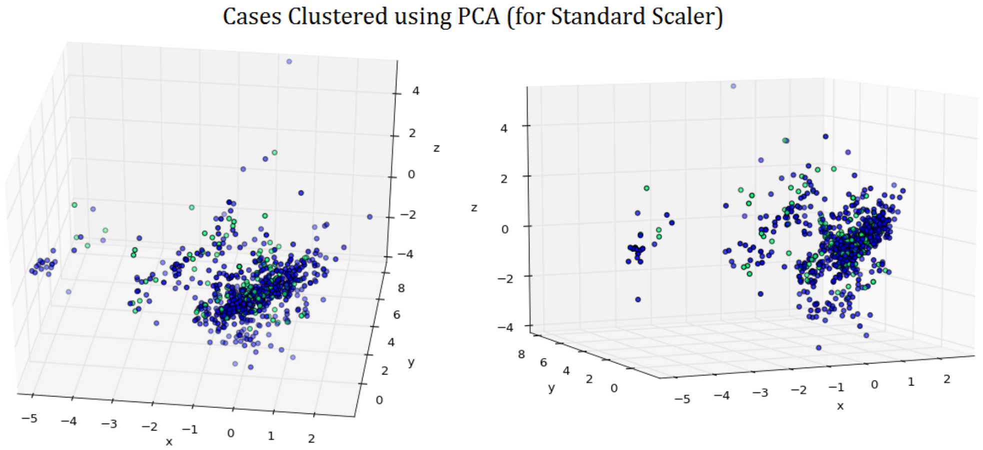


Figure 9: Clustering of cases using standard scaler, from two perspectives

When the sklearn.preprocessing.StandardScaler was used, each attribute was scaled so that it had a mean of 0 and a variance of 1. This method of normalisation was not very effective. For visualisation purposes for the report, the number of dimensions was reduced from 9 to 3. From the clustering of cases in Figure 9, there was a main cluster where most of the cases are located, and another smaller cluster. When only 3 dimensions were used, only 37% of the information was retained. This is evident from the poor clustering when the standard scaler was used.

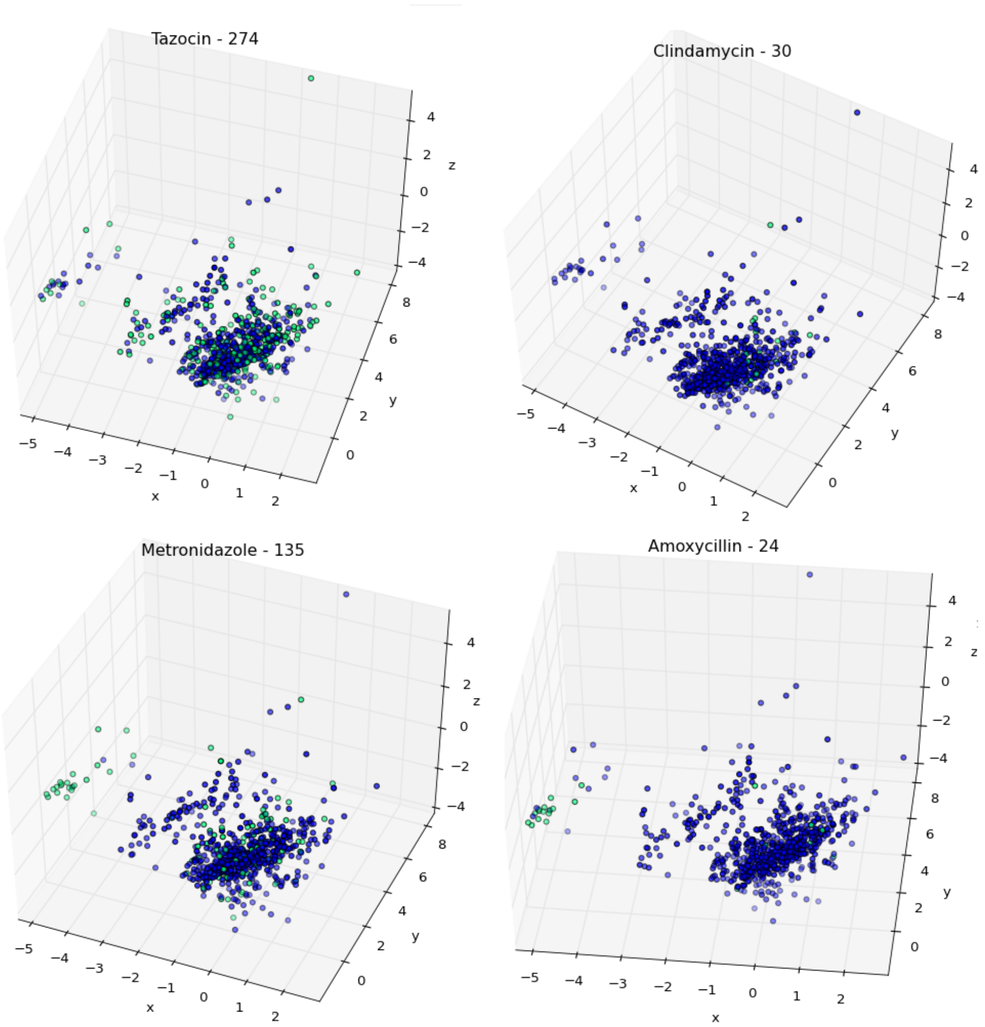


Figure 10: Clusters observed when PCA was applied using standard scaler. Cases when (a) Tazocin, (b) Clindamycin, (c) Metronidazole, and (d) Amoxycillin were prescribed are in green.

The full results for all the drugs is given in Appendix B.1. The results for Tazocin, Clindamycin, Metronidazole, and Amoxycillin are used as examples and are shown in Figure 10. It can be seen that wide-spectrum drugs like Tazocin were not well clustered. In contrast, Clindamycin, a narrow spectrum drug, was only prescribed in cases in the right-half of the main cluster, when $x > -1$. It it interesting to note that almost all the cases in the small cluster were prescribed Metronidazole. Also, most of the cases in the lower half of the small cluster (when $y < 2$) were prescribed Amoxycillin. This shows that by using just the case descriptions, the cases can be clustered. Also, there is a relationship between the clusters and the drugs prescribed.

MinMax Scaler

The Minmax scaler normalises all the attributes to have values between 0 and 1. This method of scaling is preferred to the standard scaler because it is similar to how the attributes had been scaled in the Equals and Interval functions in Section 3.4.1. Also, when the dimensionality was reduced to $m = 3$, 61% of the information was preserved, which is better than when the standard scaler was used.

When the Minmax scaler was used with PCA, the cases could be visualised in the new 3-dimensional space, shown in Figure 11. There were more obvious clusters as compared to when the standard scaler was used. Visually, it could be seen that there are four main clusters. Each of clusters can also be split into two sub-clusters each. In Figure 11, these four clusters have been colour-coded, and each of the sub-clusters have been numbered, so that each cluster can be easily identified.

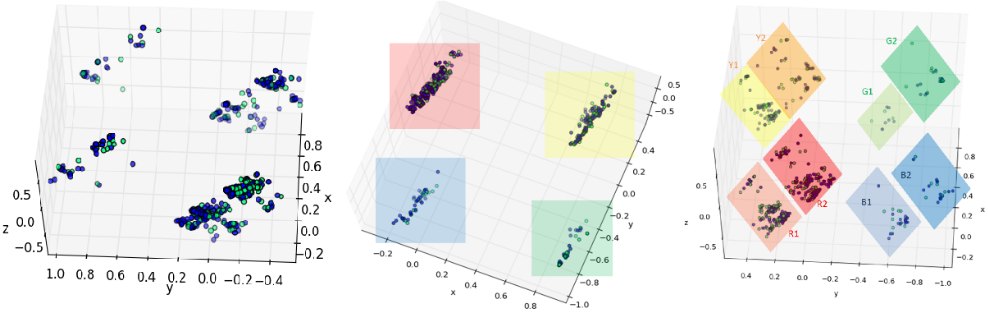


Figure 11: Clustering of cases using MinMax Scaler

In Figure 12, the cases in which the drug was prescribed is in green. Three drugs were used as examples: Tazocin, Vancomycin, and Clindamycin. For Tazocin, it is not well clustered and occurs in all eight sub-clusters. This is expected because Tazocin is a broad-spectrum antibiotic, and can be used in many cases. Clindamycin is a narrow spectrum antibiotic, and is only prescribed in cases that are in the R1, R2 and Y1 sub-clusters. For Vancomycin, since it is a narrow spectrum drug and it is expected to be well clustered like

Clindamycin. However, it can be observed that the cases in which Vancomycin were prescribed occurred in all the clusters and subclusters. This behaviour is similar to that of Tazocin and Meropenem, which are wide-spectrum drugs. This implies that Vancomycin was overpresecribed. Vancomycin was treated as a case study which is discussed in detail in Section 5.3.

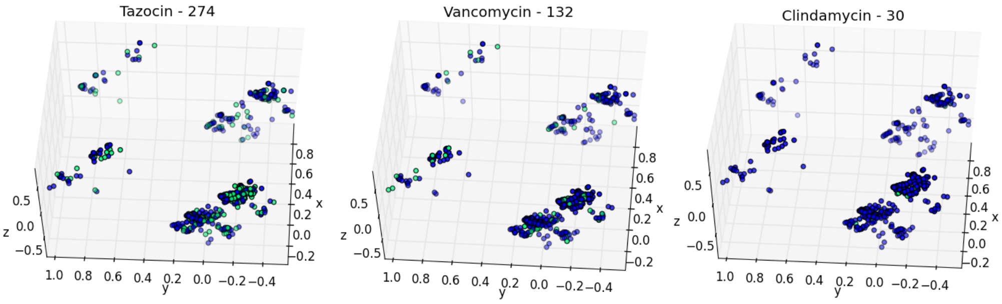
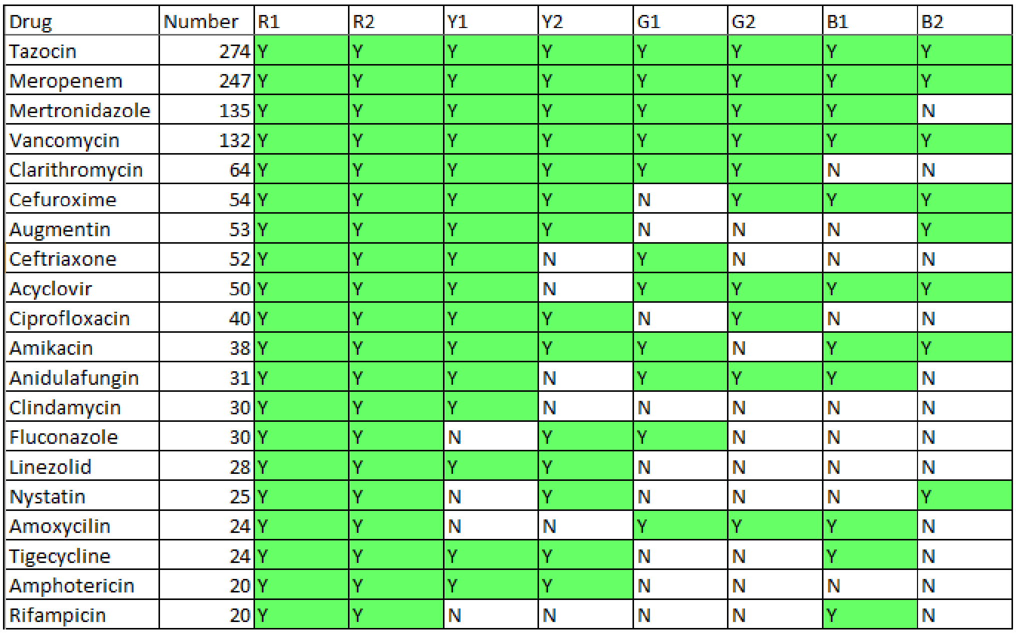


Figure 12: Clusters observed when PCA was applied using Minmax scaler. Cases when (a) Tazocin, (b) Vancomycin, and (c) Clindamycin were prescribed are in green.

The visualisation results for all the drugs is given in Appendix B.2, and the Table in Figure 13 summarises the results for all the drugs.



Drug	Number	R1	R2	Y1	Y2	G1	G2	B1	B2
Tazocin	274	Y	Y	Y	Y	Y	Y	Y	Y
Meropenem	247	Y	Y	Y	Y	Y	Y	Y	Y
Mertronidazole	135	Y	Y	Y	Y	Y	Y	Y	N
Vancomycin	132	Y	Y	Y	Y	Y	Y	Y	Y
Clarithromycin	64	Y	Y	Y	Y	Y	Y	N	N
Cefuroxime	54	Y	Y	Y	Y	N	Y	Y	Y
Augmentin	53	Y	Y	Y	Y	N	N	N	Y
Ceftriaxone	52	Y	Y	Y	N	Y	N	N	N
Acyclovir	50	Y	Y	Y	N	Y	Y	Y	Y
Ciprofloxacin	40	Y	Y	Y	Y	N	Y	N	N
Amikacin	38	Y	Y	Y	Y	Y	N	Y	Y
Anidulafungin	31	Y	Y	Y	N	Y	Y	Y	N
Clindamycin	30	Y	Y	Y	N	N	N	N	N
Fluconazole	30	Y	Y	N	Y	Y	N	N	N
Linezolid	28	Y	Y	Y	Y	N	N	N	N
Nystatin	25	Y	Y	N	Y	N	N	N	Y
Amoxycilin	24	Y	Y	N	N	Y	Y	Y	N
Tigecycline	24	Y	Y	Y	Y	N	N	Y	N
Amphotericin	20	Y	Y	Y	Y	N	N	N	N
Rifampicin	20	Y	Y	N	N	N	N	Y	N

Figure 13: Summary of clusters which had cases which used that drug (Y: Yes, N: No)

From the table, it can be seen that Tazocin, Meropenem and Vancomycin were poorly clustered and were prescribed in all 8 clusters. The red clusters (R1 and R2) contained most of the cases, and in the red cluster, any one of the drugs could have been prescribed. In general, there seems to be an inverse relationship between the number of times a drug was prescribed, and how well clustered it was. The relationship between how the clustering of a drug affects its prediction accuracy is discussed in Section 4.3.

Because the axes used in PCA are the eigenvectors of the data and have no physical meaning, it is hard to interpret what the cases in the same cluster had common. However, from the PCA results, it could be concluded there is some relationship between the attributes in a case's description, and what drug it was prescribed. This means that the data was suitable for CBR. However, because some drugs were not well clustered, the IDSS may not be able to predict those drugs very accurately.

4.2 Accuracy of Prediction

The performance of the IDSS was measured in terms of how well the system was able to model the decision making process of an expert. The suggested solution was compared to the actual solution given by an expert, and its accuracy was measured. The study was first done on cases where only one drug was prescribed per patient. The study was then extended to all cases, regardless of the number of drugs prescribed per case.

4.2.1 *Cases with only one drug prescribed*

By filtering the data using the method in Section 3.1.2, only cases where a single drug was prescribed per case were chosen to form the case base. This was chosen as the starting point of the investigation to determine the best choice of some parameters before proceeding to test using the entire case base. For example, this allows a comparison between multiclass and multilabel accuracy. For the trials in this section, an arbitrary $k = 5$ neighbours were used; the optimal number of neighbours will be determined in Section 4.4.

It is interesting to note that out of 275 cases where only one drug is prescribed, most of the cases were prescribed Tazocin (41%) and Meropenem (23%). For the remaining cases, some drugs were only prescribed rarely. Thus, a lower threshold T_L was set so that if a drug is prescribed too few times, that drug is removed from the case base. The threshold was initially set at 30, which means that a drug had to be prescribed at least 30 times in the case base. At $T_L = 30$, there were only two possible drugs in the case base: Tazocin and Meropenem. T_L was varied over different trials to investigate how the accuracy changes when there are more possible drugs. The results are shown in Table 7 for both multiclass and multilabel classification.

An upper threshold T_H was also set to remove drugs that occur too frequently. The threshold was set at 1/3 of the number of cases in the case base i.e. if a drug was prescribed in more than 1/3 of the cases, it was removed. The effect of removing this bias is shown in Trials 3 and 4 of Table 7.

In Table 7, for the "Basic" case, all the attributes are assigned equal weights. The "Best" average accuracy per case is obtained when the optimal set of weights is applied.

Trial	No. of cases tested	No. of Drugs Tested	Multiclass		Multilabel	
			Basic	Best	Basic	Best
1.	175	2	56.3%	67.5%	54.6%	66.4%
2.	202	3	47.4%	56.3%	49.6%	57.8%
3.	105	3 (no bias)	62.9%	67.4%	67.7%	68.3%
4.	130	6 (no bias)	47.7%	59.5%	56.4%	66.6%
5.	243	7	37.4%	47.4%	44.9%	45.5%

Table 7: Average accuracy per case for multiclass and multilabel classification

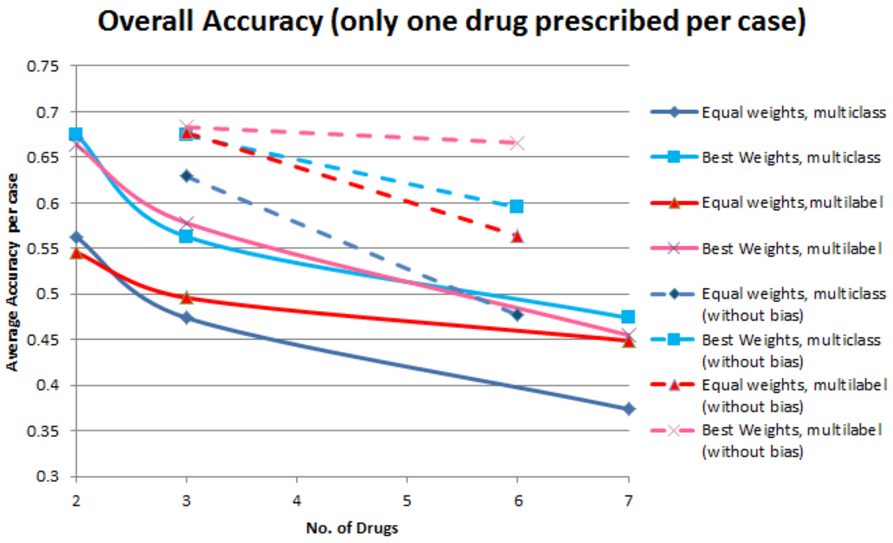


Figure 14: Overall accuracy per case for multiclass and multilabel classification

The results in Table 7 was complied into a graph shown in Figure 14. From Figure 14, some trends were observed.

1. When the "Best" set of weights was used, there is clearly an improvement in the results as compared to when equal weights were used. The improvement as particularly significant when multiclass classification is used. This indicates that the method of weighing attributes helps to better model the decision making process of an expert.

2. By setting T_H to remove drugs that occur in more than a third of the cases, the accuracy was significantly improved.

3. In general, when the number of drugs increase,the average accuracy per case decreased. This is expected because as the number possible drugs increases, the difficulty of the problem increases. For example, when there were only 2 drugs, a random guess would give an accuracy of 50%. Thus, the IDSS improved the accuracy by 1.3 times. The improvement in accuracy, as compared to a random guess, is shown in Figure 15.

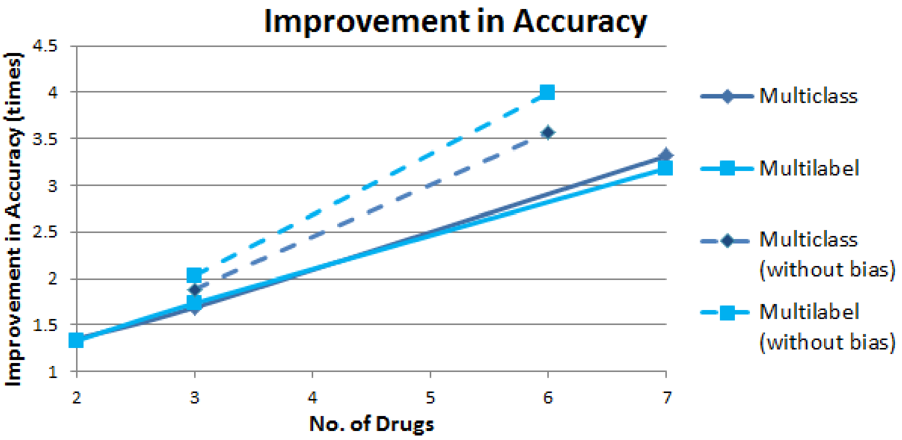


Figure 15: Improvement in accuracy as compared to a random guess

It can be seen that the improvement in accuracy increases as the number of drugs increases. This is because the accuracy was approximately constant at about 50%, even when the difficulty of the problem increased. As before, the IDSS performs better when the bias was removed.

One possible reason why the improvement in accuracy was not very significant when only two drugs are predicted was because of the nature of the drugs considered: Tazocin and Meropenem. From Section 4.1, Tazocin and Meropenem are very poorly clustered. Thus, it is difficult to differentiate between Tazocin and Meropenem. This concept is explained further in Section 4.3.

Weights

The "Best" set of weights was different for each of the trials in Table 7. This is because a different set of drugs was tested in each trial, and each drug had its own optimal set of weights. This is further discussed in Section 4.5 and Section 5.2.

Multiclass vs. Multilabel Classifier

From both Figure 14 and Figure 15, the multilabel classifier performed at least as well as the multiclass classifier for any number of drugs. The multilabel classifier also performed significantly better than the multiclass classifier when the bias was removed from the data. Furthermore, the multiclass classifier was not suitable when a case could have more than drug prescribed. Thus, the multilabel classifier was chosen for the final model, regardless of how many drugs are prescribed per case.

Removing Bias

It was hypothesized that if a drug occurred too frequently, it would bias the data. The impact of the bias is also closely linked to the number of neighbours chosen. This is because if k was too large, majority of the k neighbours would be of that drug. Thus, for all subsequent tests, T_H is set at $1/3$ of the case base and if a drug is present in more than $1/3$ of the cases, it is removed.

4.2.2 *Cases with one or more drugs prescribed*

The IDSS was tested to predict cases where one or more drugs were prescribed per case. This was done using multilabel classification. The lower threshold was set to $T_L = 30$ and the upper threshold T_H is $1/3$ of the case base. There were a total of 466 cases in the case base, and 12 possible drugs. Because there can be more than one drug prescribed at a time, there are 4095 possible combinations of drugs. Realistically, not all combinations of drugs are prescribed. From the data, there were 239 unique combination of drugs. The performance of the system was measured with respect to the overall accuracy per case and the accuracy per drug.

Overall Accuracy per Case

The average accuracy per case shown in Figure 16 compares the results with and without forced prediction. When equal weights were assigned to the attributes, the average accuracy per case was 18.50% without forced prediction, and 19.91% with forced prediction. Although this is lower than the accuracy when prescribing only a single drug per case, this is because of the significant increase in the difficulty of the problem. If out of the 239 possible combinations a random guess is made, it would only have an accuracy of 0.4%. Then, the IDSS actually improved the accuracy of prediction significantly. Furthermore, when optimal weights were used, the accuracy improves and the average accuracy is 25.0% per case without forced predictions and 25.2% with forced predictions. This shows that by assigning the weights optimally, the accuracy of the prediction was improved. When there was a forced prediction as described in Section 3.5.2, the accuracy improved further. The accuracy increased significantly when equal weights were used, and slightly when the optimal weights are used. The impact of forcing predictions is more apparent in the calculation of accuracy per drug.

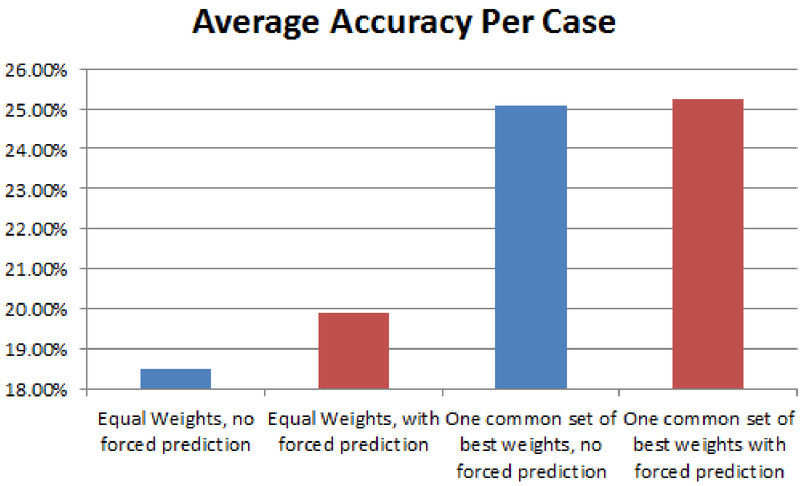


Figure 16: Average accuracy per case, when one or more drugs can be prescribed per case

It should be noted that the term ”accuracy” is used very loosely here. An ”accuracy” of 25.2% indicates that approximately one out of four times, the expert prescribes the same drug as that recommended by the IDSS. However, it could be possible that even if an expert was given the same case twice on two separate occasions, he might not suggest exactly the same drugs. This could be because some drugs are very similar and interchangeable, or simply because of human error. This would then affect the data as similar cases were not always prescribed the same drugs.

Accuracy for each Drug

The accuracy of each drug was measured in terms of sensitivity, specificity and LR+, which was calculated using Equations 6, 7 and 8 respectively. The overall best weights was the set of weights that gave the highest average accuracy per case. This same set of optimal weights was applied to all the drugs. The accuracy when using the overall best weights was compared to assigning equal weights to each attribute. Furthermore, each drug had a different optimal set of weights corresponding to it. This was the set of weights that corresponded to the highest LR+ for that drug. Note that for the accuracy of the drug-specific best weights given in Tables 8 to 10, there was also forced predictions.

Sensitivity

Drug	No. of Instances of Drug	Using Equal Weights		Using Overall Best Weights		Using Drug's Best Weights
		Not Forced	Forced	Not Forced	Forced	
Metronidazole	135	36.59%	34.37%	44.30%	45.63%	48.44%
Vancomycin	132	29.85%	29.70%	38.79%	40.61%	38.79%
Clarithromycin	64	16.25%	11.88%	43.13%	45.63%	22.19%
Cefuroxime	54	22.59%	23.33%	12.22%	14.44%	28.52%
Augmentin	53	6.79%	6.04%	18.87%	22.64%	22.26%
Ceftriaxone	52	21.92%	23.08%	38.08%	37.69%	39.23%
Acyclovir	50	11.20%	11.20%	26.80%	28.40%	23.20%
Ciprofloxacin	40	12.50%	13.00%	18.00%	20.00%	15.00%
Amikacin	38	5.26%	5.26%	7.37%	6.32%	11.58%
Anidulafungin	31	16.77%	15.48%	53.55%	61.29%	49.68%
Clindamycin	30	22.00%	22.67%	28.00%	30.67%	34.00%
Fluconazole	30	11.33%	11.33%	18.67%	26.00%	21.33%

Table 8: Drug's Sensitivity (TPR) when using equal, overall best and drug-specific best weights, with and without forced prediction

Table 8 shows the sensitivity of the test, which is calculated using the True Positive Rate (TPR). A sensitivity of x% means that if a drug is prescribed by the doctor, x is how likely the drug was also suggested by the IDSS. From Figure 17, it is evident that when the drug-specific optimal set of weights were used, the sensitivity is higher than when equal weights are allocated. This is true for all drugs. Thus, the optimal weight assignment is effective in increasing the sensitivity of the drug.

When the set of overall best weights was used, the accuracy for most of the drugs is higher than when all the attributes are weighted equally. For some drugs, they do not perform better when the overall best set of weights is used is because each drug has its own set of weights. For example, from Figure 23, 'Chest Examination' is an important attribute for Cefuroxime, but it is given a weight of 0 in the overall best set of weights.

The results also compares the situation when a prediction is forced compared to when it is not forced. it can be seen in Table 8 that by forcing a predic-

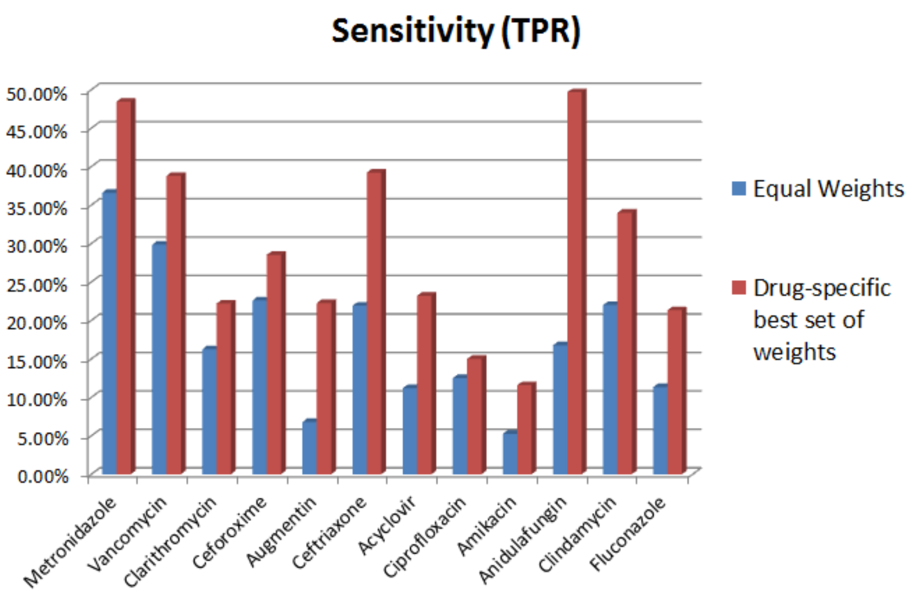


Figure 17: Best set of weights for each drug

tion, the sensitivity increases for most drugs. This is especially true when the overall best weights are used, and 10 out of the 12 drugs have a higher TPR when a prediction is forced. This is expected because if a prediction is not made, the number of False Negatives (FN) would automatically increase, and correspondingly the TPR will decrease. Thus, by forcing predictions, the TPR increases.

Specificity

The IDSS performs very well with regards to specificity. From Table 9, the average specificity over all the drugs is 92.6% when equal weights were used, and 94.2% when the best weights for each drug were used. A high specificity is very important because it means that if a drug is not proposed by the IDSS, the likelihood that this is an accurate suggestion is approximately 94%. This would help in preventing the prescription of superfluous antimicrobial drugs, thus helping tackle antimicrobial resistance.

For 8 out of 12 of the drugs, when the drug-specific best set of weights was used, the specificity is higher than when the attributes were weighed equally. The specificity does not improve for all drugs because of the trade-off between sensitivity and specificity. Thus, when the drug-specific optimal weights were used, the specificity may have been compromised for a higher sensitivity. The LR+ is calculated to measure the trade off between sensitivity and specificity.

Drug	No. of Instances of Drug	Using Equal Weights		Using Overall Best Weights		Using Drug's Best Weights
		Not Forced	Forced	Not Forced	Forced	
Metronidazole	135	84.29%	83.50%	85.86%	85.68%	88.10%
Vancomycin	132	85.93%	84.61%	82.81%	81.02%	89.10%
Clarithromycin	64	98.06%	98.16%	95.07%	94.93%	97.91%
Cefuroxime	54	94.85%	95.19%	95.39%	94.81%	96.60%
Augmentin	53	96.13%	95.84%	94.96%	94.67%	96.27%
Ceftriaxone	52	96.86%	96.86%	96.38%	96.04%	96.38%
Acyclovir	50	97.88%	97.45%	96.78%	96.68%	97.79%
Ciprofloxacin	40	97.98%	97.89%	97.00%	96.57%	98.12%
Amikacin	38	97.57%	97.80%	97.20%	97.50%	98.32%
Anidulafungin	31	98.90%	99.03%	96.87%	96.64%	98.34%
Clindamycin	30	96.56%	99.56%	98.03%	98.21%	98.49%
Fluconazole	30	97.52%	96.93%	96.97%	96.65%	98.94%

Table 9: Drug's Specificity (TNR) when using equal, overall best and drug-specific best weights, with and without forced prediction

When predictions are forced, the specificity worsens for 10 out of 12 drugs. This is expected because when predictions are forced, the number of False Positives would increase. Correspondingly, the TNR decreases. Again, LR+ is computed to measure the trade off between sensitivity and specificity with regards to forcing predictions.

Positive Likelihood Ratio (LR+)

LR+ quantifies the trade-off between sensitivity and specificity and the LR+ results are shown in Table 10. An LR+ value of 1 indicates a baseline performance (where True Positive Rate = False Positive Rate), and the higher the LR+, the higher the confidence of the system. In general, the IDSS performs very well with regards to specificity. From Figure 18, it can be seen that by using the best weights for each drug, the LR+ for each drug was higher than when the overall best set of weights or when a set of equal weights was used.

For 9 out of 12 drugs, when the overall best set of weights is used, the LR+ improves as compared to when equal weights are used. For Cefuroxime, Ciprofloxacin and Amikacin, the LR+ does not improve when the overall best set of weights is used. This is probably because the attributes that are important in these drugs like 'Respiratory Rate' have a weight of 0 in the overall best set of weights. Thus, the overall best set of weights was not suitable for prescribing these drugs.

Drug	No. of Instances of Drug	Using Equal Weights		Using Overall Best Weights		Using Drug’s Best Weights
		Not Forced	Forced	Not Forced	Forced	
Metronidazole	135	2.33	2.08	3.13	3.19	4.07
Vancomycin	132	2.12	1.93	2.26	2.14	3.56
Clarithromycin	64	8.38	6.45	8.76	8.99	10.62
Cefuroxime	54	4.39	4.86	2.65	2.78	8.39
Augmentin	53	1.75	1.45	3.75	4.25	5.97
Ceftriaxone	52	6.98	7.35	10.51	9.52	10.83
Acyclovir	50	5.29	4.40	8.32	8.56	10.49
Ciprofloxacin	40	6.19	6.15	5.99	5.84	7.99
Amikacin	38	2.17	2.40	2.63	1.80	6.88
Anidulafungin	31	15.20	16.04	17.13	18.26	30.01
Clindamycin	30	6.39	6.59	14.20	17.14	22.46
Fluconazole	30	3.58	3.69	6.17	7.76	20.22

Table 10: Drug’s Positive Likelihood Ratio (LR+) when using equal, overall best and drug-specific best weights, with and without forced prediction

From Table 8, it can be observed that when predictions are forced, the increase in TPR outweighs the reduction in TNR. For 8 out of the 12 drugs, the LR+ improved when a forced prediction was used together with the overall best set of weights. So, when finding the optimal set of weights for each specific drug, predictions were always forced.

4.3 Relationship between Clustering and Accuracy

On the surface, there seems to be a relationship between how frequently a drug is prescribed, and its accuracy in terms of LR+. From Table 10, the two most frequently prescribed drugs, Metronidazole and Vancomycin, are prescribed very frequently and have a poor LR+. Anidulafungin, Clindamycin and Fluconazole are rarely prescribed and have very high LR+. However, this trend is not true because drugs like Amikacin and Ciprofloxacin are also infrequently prescribed but have a low LR+.

From Figure 19, it can be observed that there is a general trend between how well clustered a drug is, which is quantified by the number of clusters the drug occurs in, and its accuracy, which is measured using LR+. Generally, if the drug is more well clustered, the drug’s accuracy increases. For example, Amikacin is a broad-spectrum drug that occurs in all 8 sub-clusters. As such, it was not well modelled in the IDSS and therefore has a low LR+ of 6.88. In

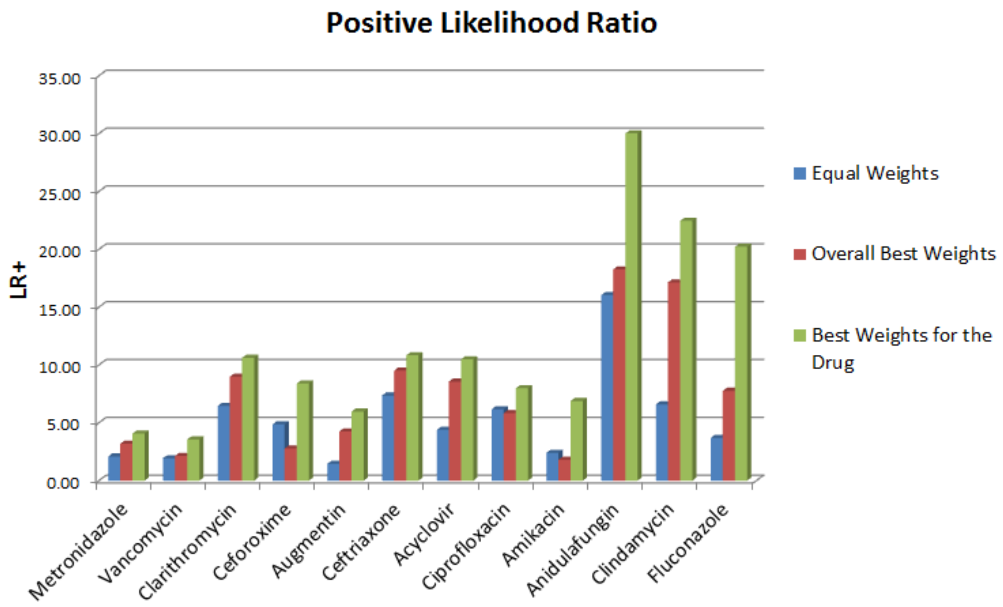


Figure 18: Comparing LR+ for equal weights, overall best weights, and drug-specific best weights

contrast, Clindamycin is a narrow-spectrum drug that is well clustered, as seen in Figure 12. Thus, it was modelled well in the IDSS and it has a high LR+ of 22.46.

It can also be observed that when equal weights are used, this gives the lowest accuracy. On the other hand, drug-specific weights gave the highest accuracy.

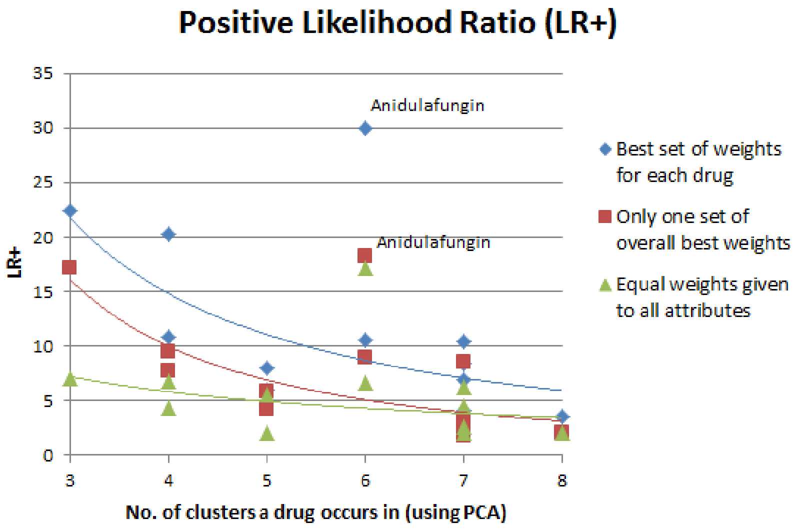


Figure 19: Relationship between clustering and Positive Likelihood Ratio (LR+)

In Figure 19, there seemed to be an outlier that corresponds to Anidulafungin. For the case of Anidulafungin, it was observed that although it occurs in 6 of the 8 sub-clusters, most of the cases generally fell into only one cluster, which has two sub-clusters. Thus, it can be considered a well-clustered drug, and therefore has a high LR+.

4.4 Choosing the Number of Neighbours, K

In order to decide how many neighbours, K, was the most appropriate for the IDSS, the overall accuracy was measured when K was varied between 3 and 15. In the tests, T_L was set to 30 and T_H was set to $1/3$ of the number of cases. The "Forced Prediction" configuration is used. The results are shown in Table 11.

K	No. of Cases	Avg. No. of Predictions Made	Avg. No. of Correct Predictions	Accuracy	
				A_1:Exclude No Prediction	A_2:Include No Prediction
3	466	411.4	85.2	20.71%	18.28%
5	466	371.8	84.8	22.81%	18.20%
7	466	327.2	79.6	24.33%	17.08%
15	466	255.8	67.6	26.43%	14.51%

Table 11: Accuracy of predictions as the number of neighbours, K, is varied

Firstly, K was chosen to be an odd number so that there is a clear majority. Also, K had to be smaller than twice of T_L, especially if there is no forced predictions. This is because if $K > 2 \times T_L$, for a drug that is only prescribed T_L times, it would never be predicted as it would not have more than half of its neighbouring cases being prescribed that drug.

From the results in Table 11, it can be observed that the number of predictions is less than the total number of cases in the case base. When K was increased, the average number of predictions decreased. This is expected because there are more neighbours and there may be no clear majority, even if a prediction was forced. Secondly, the average number of correct predictions decreases when K increases. This is probably because most of the drugs are not well clustered, thus many of the neighbours that are further from the testing case may have been prescribed a different drug than that of the testing case. A_1 and A_2 were two measures of accuracy that were calculated using Equations 9 and 10 respectively. The objective was to have A_1 and A_2 as high as possible.

$$A_1 = \frac{No.\ of\ correct\ predictions}{No.\ of\ predictions} \tag{9}$$

$$A_2 = \frac{No.\ of\ correct\ predictions}{Total\ no.\ of\ cases} \tag{10}$$

From Figure 20, it is evident that A_1 is directly proportional to K, while A_2 is inversely proportional to K. K should be chosen to maximise both A_1 and A_2. To measure the trade-off between A_1 and A_2, a simple trade-off equation is used (Equation 11).

$$T = \frac{A_2}{1 - A_1} \tag{11}$$

From the equation, T will increase when A_2 increases, and T will also increase if A_1 increases. T is plotted against K in Figure 21

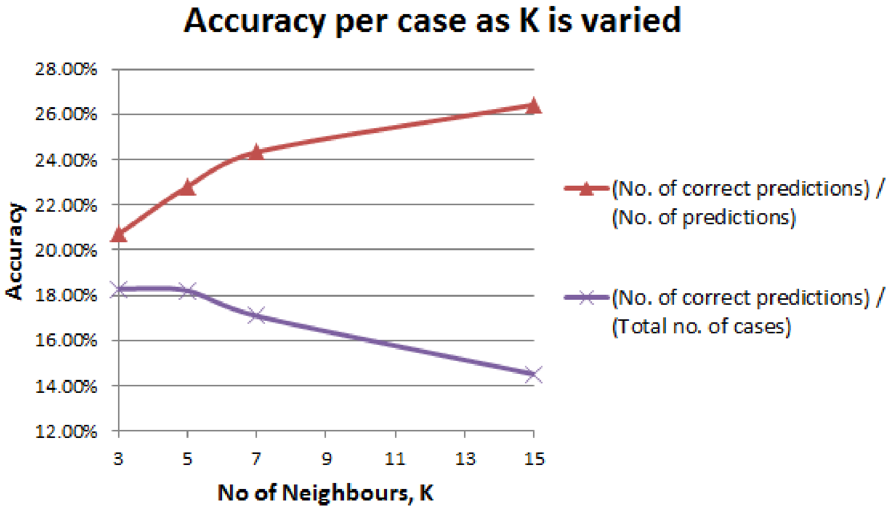


Figure 20: Accuracy A_1 and A_2 when number of neighbours, K, is varied

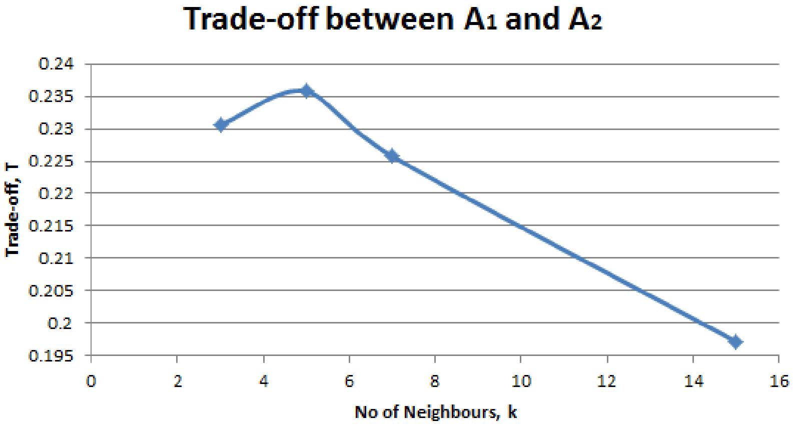


Figure 21: Trade-off, T, when number of neighbours, K, is varied

From Figure 21, K is chosen to be 5 because T was maximised at that point. However, it should be noted that the optimal number of neighbours depends on how the doctors prefer the IDSS to be configured. If they prefer to have more correct predictions in general, then K should be reduced to maximise A_2. In contrast, if the objective is that among the predictions, there should be a large number of correct predictions, then K should be increased to maximise A_1.

4.5 Weights Allocation

The overall best set of weights is the set of weights that gave the highest average accuracy per case. This section investigates the relationship between the overall best set of weights and the best set of weights for each drug.

4.5.1 *Overall best set of weights*

Figure 22 shows the set of weights that was used to achieve the highest average accuracy per case, as described in Section 4.2.2. The weights have a precision of ±2.5%. These weights mimic the relative importance an expert places on each attribute when deciding which drug(s) to prescribe.

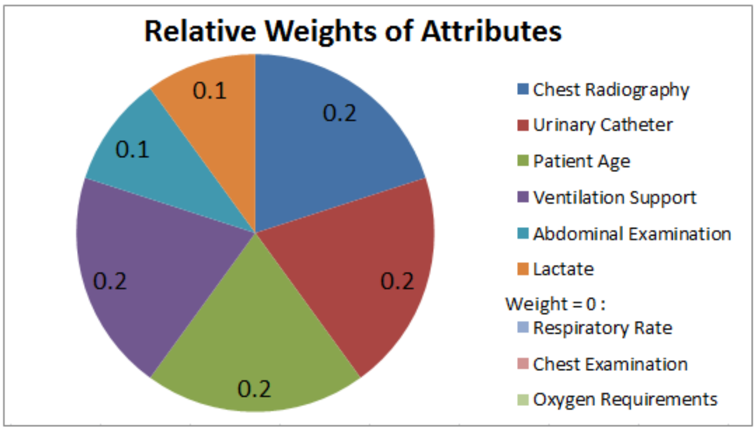
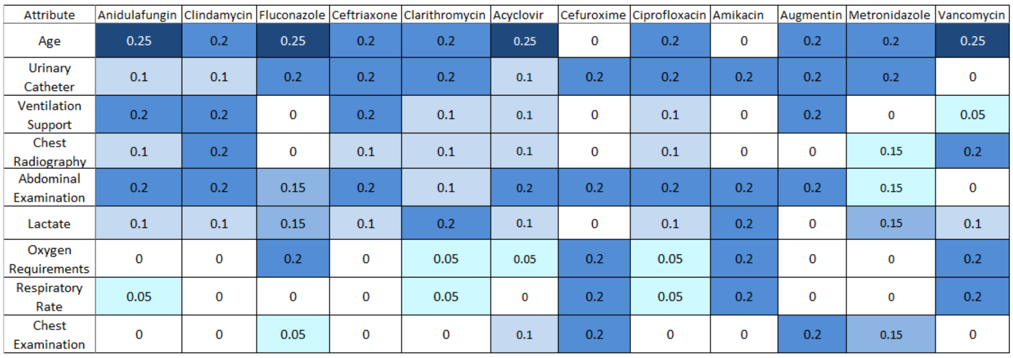


Figure 22: Relative weights of attributes for overall best accuracy per case

In a qualitative study done by Kushniruk, Patel and Fleiszer (1995), results showed that experts (intensive care specialists) usually came up with more refined approaches than intermediates (intensive care residents). In particular, experts tend to focus less on laboratory tests and more on the patient's situation, as compared to intermediates. For example, experts would request for information such as the patient's history and if the patient had any problems when leaving the operating theatre. [30] These observations are consistent with the weights observed in Figure 22.

4.5.2 *Optimal weights for each drug*



Attribute	Anidulafungin	Clindamycin	Fluconazole	Ceftriaxone	Clarithromycin	Acyclovir	Cefuroxime	Ciprofloxacin	Amikacin	Augmentin	Metronidazole	Vancomycin
Age	0.25	0.2	0.25	0.2	0.2	0.25	0	0.2	0	0.2	0.2	0.25
Urinary Catheter	0.1	0.1	0.2	0.2	0.2	0.1	0.2	0.2	0.2	0.2	0.2	0
Ventilation Support	0.2	0.2	0	0.2	0.1	0.1	0	0.1	0	0.2	0	0.05
Chest Radiography	0.1	0.2	0	0.1	0.1	0.1	0	0.1	0	0	0.15	0.2
Abdominal Examination	0.2	0.2	0.15	0.2	0.1	0.2	0.2	0.2	0.2	0.2	0.15	0
Lactate	0.1	0.1	0.15	0.1	0.2	0.1	0	0.1	0.2	0	0.15	0.1
Oxygen Requirements	0	0	0.2	0	0.05	0.05	0.2	0.05	0.2	0	0	0.2
Respiratory Rate	0.05	0	0	0	0.05	0	0.2	0.05	0.2	0	0	0.2
Chest Examination	0	0	0.05	0	0	0.1	0.2	0	0	0.2	0.15	0

Figure 23: Best set of weights for each drug

In Figure 23, the weights assigned to each attribute has been colour-coded: the higher the weights, the darker the colour. Some attributes such as 'Urinary Catheter', 'Abdominal Examination' and 'Age' are commonly allocated high weights for almost all the drugs. On the other hand, some attributes like 'Chest Examination' is allocated only a small weight for most of the drugs. This is consistent with the overall best set of weights in Figure 22. Quantitatively, in Table 12, the overall best weights is compared to the weights of the attributes averaged over all the drugs. Also, a weighted average was considered, with drugs prescribed more frequently being given higher weights.

Attribute	Overall Best	Average	Weighted Average
Patient Age	0.2	0.17	0.191
Urinary Catheter	0.2	0.16	0.147
Ventilation Support	0.2	0.07	0.078
Chest Radiography	0.2	0.09	0.108
Abdominal Examination	0.1	0.15	0.142
Lactate	0.1	0.11	0.111
Oxygen Requirements	0.0	0.09	0.083
Respiratory Rate	0.0	0.09	0.073
Chest Radiography	0.0	0.07	0.068

Table 12: Overall best weights and weights averaged from drug-specific weights

From Table 12, it can be seen that the attributes that were given a weight of 0 in the set of overall best weights also had the lowest weights for the average and weighted average case. Also, the two most important attributes according to the drug-specific weights were 'Age' and 'Urinary Catheter'. This is consistent with the overall best weights.

In Section 4.2.1, when the study was limited to only cases with one drug per patient, the best weights for each configuration was different. This is because the drugs for each combination was different. The, it is expected that the overall best set of weights would be different, depending on the drugs considered for the new case. For furture applications, if a doctor does not have any drug in mind, he may want to use the overall best weights or equal weights. However, if he wants to check if a particular drug or combination of drugs is suitable for a new case, he may want to use that drug-specific weights instead.

4.6 Final IDSS Model

This section summarises the set of parameters that were chosen for the final model, to achieve the best possible system performance. Lastly, the time complexity of the final model was considered.

4.6.1 *Configuration*

Each parameter was tested independently with all other parameters held constant to find the best configuration for each paramter. Then, the best choice for each parameter was used in the final IDSS model. The parameters are as follows:

- Removing bias from the case base by setting T_H to one third of the number of cases in the case base.

- Multilabel Classification instead multiclass classification.

- Forcing prediction of a drug so that as long as it has the highest prediction probability among all the drugs, even if the drug is not prescribed in more than 50% of the case's neighbours, that drug will still be suggested.

- Choose number of neighbours, $K = 5$.

- Weight set to best overall set of weights described in Figure 22. However, if only certain drugs are being considered, may choose to use the drug-specific set of best weights instead.

Using these parameters, the IDSS is able to model the decisions of an expert with a minimum average accuracy per case of 25.2%. This is the accuracy for "worst-case-scenario" which assumes that if the drug suggested by the IDSS is different from that which had been prescribed by an expert, then the prediction is wrong. Actually, antibiotics fall into there are 15 possible classes, and antibiotics within the same class are similar. For example, both Clindamycin and Clarithromycin are of the Macrolides class and can be considered similar. Some other drugs may also be used interchangeably for common infections. [31] This similarity is not captured in the accuracy score, because it is difficult to quantify the similarity between drugs. Thus, it can be expected that the actual performance of the system is higher than estimated.

4.6.2 *Time Complexity*

As the IDSS will be implemented in the real-world, it is important that when a new case is received, the IDSS can suggest a solution without too much time delay.

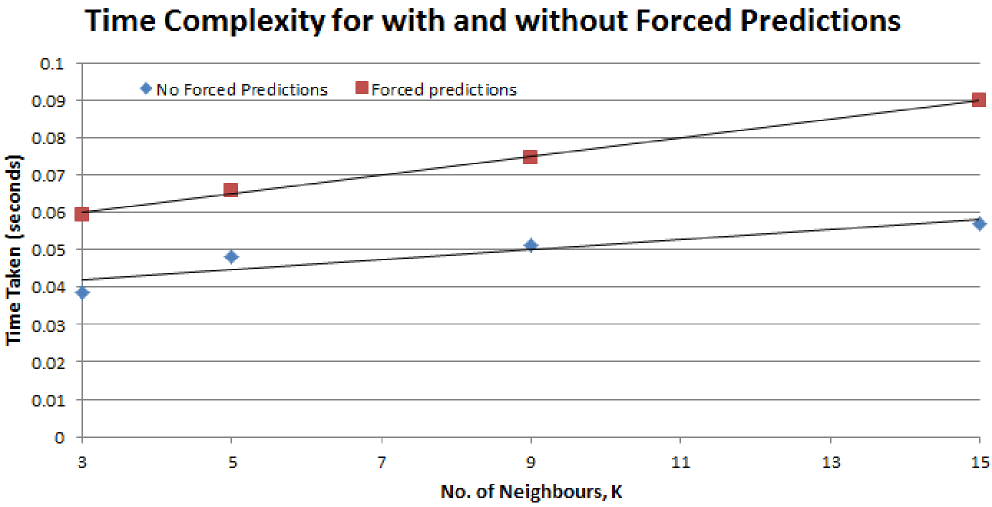


Figure 24: Time taken as the number of neighbours, *K*, is varied, with and without forced predictions.

In Figure 24, the time taken for the IDSS to make a prediction is measured. There were 130 cases used in the case base, and the number of neighbours, *K*, is varied. it can be observed that the complexity increases linearly with the number of neighbours, *K*. Also, if predictions are forced, the time taken is longer, which is expected.

In the actual implementation of the system, the only variable that would change is the number of cases in the case base. By timing the system, it was observed in Figure 25 that the time taken varied approximately linearly with the number of cases in the case base. This is expected because the complexity of the KNN algorithm is $\mathcal{O}(N)$. [32]

The time taken for the actual system is predicted in Figure 25. If the case base has $20,000$ cases, the estimated time taken for the IDSS to retrieve similar cases and make a prediction is about 30 seconds. It is expected that the case base would have around 20000 cases, and not more than $40,000$ cases. This is because cases older than 1.5 to 2 years would be removed from the case base. This is to constantly update the knowledge because the behavior of the pathogens change over time and these pathogens would mutate and develop a resistance to the antimicrobial drugs.

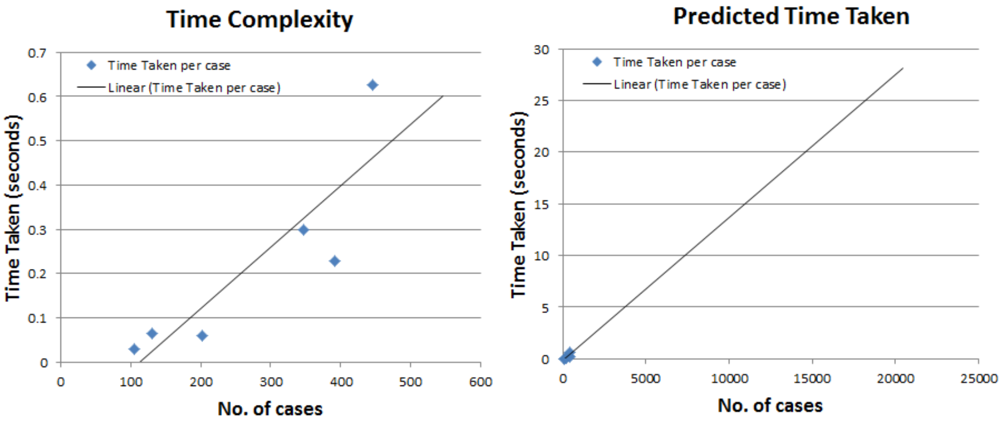


Figure 25: Time taken as the number of cases in the training set is varied. Prediction is made using a linear trend, for when the case base contains 20000 cases.

5

INTERPRETATION OF RESULTS

In this chapter, the results presented in Chapter 4 are discussed with respect to how they may be of significance to help the doctors tackle antibiotic resistance. In particular, this chapter looks at how the suggested antimicrobial treatment may be of use to doctors, the relative importance of the attributes, as well as how clustering and LR+ may be indicators of overprescription.

5.1 Evaluation of Suggested Antimicrobial Treatments

When a new case is received, the IDSS retrieves the most similar cases and a prediction is made, if possible (Figure 4). These are presented to the doctor for consideration. Using the accuracy scores in Chapter 4, the doctor can better evaluate the usefulness of the suggested solution presented.

The TPR and FPR can be presented together with the suggested solution to aid the decision process. For example, if the overall best set of weights has been used (with forced prediction) and Anidulafungin has been recommended, the doctor knows that the TPR with respect to Anidulafungin is 61.3% (from Table 8). On the other hand, the doctor may be considering whether to prescribe Amikacin for a new case. Using the best set of weights corresponding to Amikacin, if the IDSS recommends that Amikacin should not be prescribed, the doctor would know that the TNR is 98.3% (from Table 9).

Even for cases when a prediction cannot be made and there is no suggested solution, the $k = 5$ most similar cases are presented to the doctor. Since there is no prediction made, this implies that the cases presented have a wide variety of alternative treatments that were used for similar cases. These cases would be of interest to the doctor because he can compare the new case to these similar cases and decide if any of the previous antimicrobial therapies can be adapted for the new case. Since it is difficult to quantify the usefulness of the similar case presented when no prediction is made, the case is excluded from the accuracy calculation.

It should be noted that the IDSS is designed to mimic an expert's decision. There is no "correct prescription", especially for the patients in ICU where the cases are very complex. Thus, the accuracy measure is based on how well the system mimics the decision making process of an expert, and not how suitable the suggestion was. In order to make a holistic decision, the case outcome of the retrieved cases must also be considered. The outcome of the case contains information such as whether the patient's condition improved or worsened after taking the prescribed antimicrobial treatment, and whether the patient was discharged from the ICU, indicating that the patient had recovered. For example, if all of the retrieved cases which are similar to the new case had been prescribed drug A, this means that drug A is a commonly used drug for this case description. However, if the similar cases indicate that most of the patients' conditions worsened after taking drug A, then the doctor might decide to prescribe an alternative antimicrobial therapy. It could also be concluded that this is a situation where drug A is frequently misused and the doctors might want to pay more attention to these cases to prevent further misuse of this drug.

The IDSS could be limited to only consider cases where the treatment was successful, so that a suggested antimicrobial treatment would be one that has been successful in many similar cases. However, such a restriction would reduce the size of the case base, which is already quite small at present. Furthermore, by showing unsuccessful cases, the doctors can identify common misuse of antimicrobial drugs, as explained in the example above.

5.2 Interpretation of Importance of Attributes

Table 13 is an extract of the table in Appendix A. Table 13 shows the relative importance for the attributes that have been considered in the IDSS. The attributes are ranked such that Patient Age has the highest importance. The relative importance has been determined by a medical expert.

In comparison to the overall best set of weights in Table 12, there is some similarity between the attributes' importance as determined by an expert, and its importance derived from the IDSS weights. In particular, 'Patient Age' is the most important attribute out of those used in the IDSS. This is indicated by both the expert and the IDSS weights. In Table 13, many of the attributes are ranked 14. Using the IDSS, there is a finer precision for the importance of these attributes.

From the ranking in Table 13, there is only one set of importance corresponding to the attributes. However, after finding the optimal weights for each drug in

Attribute	Relative Importance
Patient Age	9
Lactate	12
Urinary Catheter	14
Ventilation Support	14
Chest Radiography	14
Abdominal Examination	14
Oxygen Requirements	14
Respiration Rate	14
Chest Radiography	14

Table 13: Ranking of the importance of attributes (extracted from Appendix A

Figure 23, it was concluded each drug had its own set of optimal weights. Thus, more than being able to model the overall decision process of an expert, the IDSS also finds the relative importance of the attributes, as considered by the doctor when prescribing a particular drug (Figure 23). This may be useful in two ways:

1. For intermediates (e.g. intensive care residents): From [30], it was observed that there is a difference in the decision making process between intermediates and experts. The intermediates may use the relative importance of the attributes as an indication of the decision making process of experts, and by using the IDSS, it can help the intermediates better mimic the experts in deciding appropriate antimicrobial treatments.

2. For experts (e.g. intensive care specialist): The relative importance of the attributes as determined by the IDSS shows the experts which attributes that had placed greater emphasis on when prescribing the drug. This may highlight some attributes that the expert had been wrongly focused on. Then, the expert may use this as a feedback and adjust their decision making process accordingly.

Thus, the weights and corresponding importance of the attributes are not only inputs to the IDSS, but they also provide insights into the decision making process of an expert. This is helpful for intermediates to possibly improve their decision making process, and for experts to evaluate their current approach to prescribing specific antimicrobial drugs. By improving the decision making process, misuse of antimicrobial drugs can be prevented.

5.3 Predicting overprescription

By using the results of the clustering using PCA, and the LR+ values, the IDSS may be able to highlight overprescription of drugs. Vancomycin will be used as a case study for this. Vancomycin is a narrow spectrum drug used for treating infections caused by Gram-positive bacteria. [33] It is a frequently prescribed antimicrobial that is used in 132 out of 466 cases.

From Section 4.1, Vancomycin occurs in all 8 sub-clusters when the min-max scalar is used. This is similar to the results of Meropenem, which is also a frequently prescribed drug. However, in contrast to Vancomycin, Meropenem is an ultra-wide-spectrum drug. Amikacin is another wide-spectrum drug that occurs in all 8 sub-clusters, even though it is only prescribed very rarely (38 times in the 466 cases). In contrast, Clindamycin is a narrow spectrum drug and it is well clustered. Thus, although Vancomycin is a narrow spectrum drug, from its clustering results, it behaves like a wide-spectrum drug.

Given the positive relationship between how well clustered a drug is, and its LR+ score, it can also be observed that Vancomycin has a very low LR+ of 3.56, which is similar that of Metronidazole (4.07). In fact, Vancomycin has the lowest LR+ score out of all the drugs, when the drug-specific best weights are used. This is in contrast to Cindamycin, which is a narrow-spectrum drug and has a very high LR+ of 22.46. Thus, Vancomycin also seems to be a broad-spectrum drug when considering the LR+ score.

After checking with medical experts, it was concluded that Vancomycin had been over-prescribed in the case base. There has been several bacteria that have been developing resistance to Vancomycin. Some examples are Vancomycin Resistant Enterococci and Vancomycin intermediate and resistant Staphylococcus aureus. This could have been due to misuse and overprescription of the drug, which is evident in the results from the IDSS.

From the example of Vancomycin, if there other drugs that are supposed to be narrow band but are poorly clustered and have a low LR+, this may be an indicator that the drug is over-prescribed. If misuse of drugs can be identified early and corrected, this will help with antimicrobial resistance.

6

CONCLUSION AND FUTURE WORK

6.1 Conclusion

In order to achieve the aim of tackling antimicrobial resistance, there were three main objectives of the projects: firstly, to understand the decision making process of an expert with respect to antimicrobial drug prescriptions; secondly, to incorporate this understanding into an IDSS that is able to suggest suitable antimicrobial drugs; and thirdly, to measure the performance of the IDSS. All three objectives were met.
To achieve the first objective, PCA was used to reduce the dimensionality of the problem. The results showed that cases could be clustered according to their case descriptions, and there is a relationship between the clusters of cases and the antimicrobial drugs prescribed. This meant that the data was suitable to be modelled using CBR.

The main objective of the project was to develop an IDSS that, when given a new case, is able to suggest suitable antimicrobial therapy options in a way that mimics the decision making process of an expert. This was achieved by designing the IDSS using the technique of CBR. In particular, KNN was used as the retrieval method, where the nearest neighbours were found using a weighted average distance metric. By using the weights to represent the relative importance of attributes, the decision making process of the expert can be better modelled. This method of weighing each attribute according to their importance was a novel application that had not been implemented in any existing systems.

Finally, the performance of the IDSS was measured in terms of how well the IDSS modelled the expert's decision making process using two main performance metrics: average accuracy per case and accuracy per drug. The high specificity of at least 84% for all the drugs indicate that the concept of CBR and KNN is suitable. Furthermore, when the optimal weights are chosen, the overall accuracy per case improves, and the sensitivity improves as well. This shows that the novel approach of weighing the attributes improves the performance of the IDSS.

By modelling the decision making process of an expert, the IDSS is not only able to suggest possible antimicrobial therapies, but it also highlights any possible mistakes in the current decision making process. For example, if a narrow-spectrum drug like Vancomycin is frequently prescribed, not well clustered and has a low LR+, this may indicate that the drug is over-prescribed. Also, if the experts do not agree with the relative importance of the attributes as determined by the IDSS, this might highlight some attributes that the expert had been overemphasizing, resulting in a misuse of the drug. Hence, the IDSS model is able to highlight misuse of antimicrobial drugs quickly, thus helping to tackle the problem of antimicrobial resistance.

6.2 Future work

The next step for the IDSS would be to perform clinical trials. This would allow the experts to evaluate the usefulness of the IDSS's suggestions. Based on the feedback, the IDSS can adjust the parameters like attributes' weights to improve the retrieval and prediction algorithm of the IDSS.

For the scope of this project, the focus of the IDSS was to suggest suitable antimicrobial drugs. As an extension, the IDSS could also be designed to be able to suggest a dosage for the drug that is customised to the demographics of the patient. Such a system would be useful even after the pathology and sensitivity information for the case is available.

BIBLIOGRAPHY

[1] European Centre for Disease Prevention and Control (ECDC). Factsheet for general public. http://ecdc.europa.eu/en/eaad/antibiotics-get-informed/factsheets/Pages/ general-public.aspx. Accessed on 17 Aug 2015.

[2] The NHS in England NHS Choices. Checklist for core elements of hospital antibiotic stewardship programs. http://www.nhs.uk/NHSEngland/ARC/Pages/AboutARC.aspx. Accessed on 15 Aug 2015.

[3] European Centre for Disease Prevention and Control. Antimicrobial resistance surveillance in europe 2013. *Annual Report of the European Antibicrobial Resistance Surveillance Network (EARS-Net)*, 2014.

[4] C. Liu, A. Bayer, S. E. Cosgrove, R. S. Daum, S. K. Fridkin, R. J. Gorwitz, S. L. Kaplan, A. W. Karchmer, D. P. Levine, B. E. Murray, M. J Rybak, D. A. Talan, and H. F. Chambers. Clinical practice guidelines by the infectious diseases society of america for the treatment of methicillin-resistant Staphylococcus aureus infections in adults and children. *Clinical Infectious Diseases*, 52(3):18–55, Feb 2011.

[5] S. K. Fridkin. Vancomycin-intermediate and -resistant Staphylococcus aureus: what the infectious disease specialist needs to know. *Clinical Infectious Diseases*, 32(1):108–115, Jan 2001.

[6] George M Eliopoulos, Sara E Cosgrove, and Yehuda Carmeli. The impact of antimicrobial resistance on health and economic outcomes. *Clinical Infectious Diseases*, 36(11):1433–1437, 2003.

[7] BBC. Antibiotic resistance: 80,000 'might die' in future outbreak. http://www.bbc.co.uk/news/uk-32193606, 2015.

[8] I. C. Gyssens, P. J. van den Broek, B. J. Kullberg, Y. Hekster, and J. W. van der Meer. Optimizing antimicrobial therapy. A method for antimicrobial drug use evaluation. *Journal of Antimicrobial Chemotherapy*, 30(5):724–727, Nov 1992.

[9] Jason A Roberts, Mohd H Abdul-Aziz, Jeffrey Lipman, Johan W Mouton, Alexander A Vinks, Timothy W Felton, William W Hope, Andras Farkas, Michael N Neely, Jerome J Schentag, et al. Individualised antibiotic dosing for patients who are critically ill: challenges and potential solutions. *The Lancet Infectious Diseases*, 14(6):498–509, 2014.

[10] R. Schmidt and L. Gierl. Case-based reasoning for antibiotics therapy advice: an investigation of retrieval algorithms and prototypes. *Artificial Intelligence in Medicine*, 23(2):171–186, Oct 2001.

[11] Leonard Leibovici, Mical Paul, Anders D Nielsen, Evelina Tacconelli, and Steen Andreassen. The treat project: decision support and prediction using causal probabilistic networks. *International journal of antimicrobial agents*, 30:93–102, 2007.

[12] Agnar Aamodt and Enric Plaza. Case-based reasoning: Foundational issues, methodological variations, and system approaches. *AI communications*, 7(1):39–59, 1994.

[13] Julie Main, Tharam S Dillon, and Simon CK Shiu. A tutorial on case based reasoning. In *Soft computing in case based reasoning*, pages 1–28. Springer, 2001.

[14] Lothar Gierl, Dagmar Steffen, Dusan Ihracky, and Rainer Schmidt. Methods, architecture, evaluation and usability of a case-based antibiotics advisor. *Computer methods and programs in biomedicine*, 72(2):139–154, 2003.

[15] Centers for Disease Control and Prevention. The antibiotic awareness campaign. http://cid.oxfordjournals.org/content/42/Supplement_2/S90.full. Accessed on 15 Aug 2015.

[16] Dereck L Hunt, R Brian Haynes, Steven E Hanna, and Kristina Smith. Effects of computer-based clinical decision support systems on physician performance and patient outcomes: a systematic review. *Jama*, 280(15):1339–1346, 1998.

[17] Peter JF Lucas. Certainty-factor-like structures in bayesian belief networks. *Knowledge-based systems*, 14(7):327–335, 2001.

[18] CAM Schurink, PJF Lucas, IM Hoepelman, and MJM Bonten. Computer-assisted decision support for the diagnosis and treatment of infectious diseases in intensive care units. *The Lancet infectious diseases*, 5(5):305–312, 2005.

[19] David E Heckerman and Edward H Shortliffe. From certainty factors to belief networks. *Artificial Intelligence in Medicine*, 4(1):35–52, 1992.

[20] Leonard Leibovici, Mical Paul, Anders D Nielsen, Evelina Tacconelli, and Steen Andreassen. The treat project: decision support and prediction using causal probabilistic networks. *International journal of antimicrobial agents*, 30:93–102, 2007.

[21] Leonard Leibovici, Michal Fishman, Henrik C Schonheyder, Christian Riekehr, Brian Kristensen, Ilana Shraga, and Steen Andreassen. A causal

probabilistic network for optimal treatment of bacterial infections. *Knowledge and Data Engineering, IEEE Transactions on*, 12(4):517–528, 2000.

[22] Alec Holt, Isabelle Bichindaritz, Rainer Schmidt, and Petra Perner. Medical applications in case-based reasoning. *The Knowledge Engineering Review*, 20(03):289–292, 2005.

[23] Stefania Montani, Riccardo Bellazzi, Luigi Portinale, Giuseppe dAnnunzio, Stefano Fiocchi, and Mario Stefanelli. Diabetic patients management exploiting case-based reasoning techniques. *Computer Methods and Programs in Biomedicine*, 62(3):205–218, 2000.

[24] Tae-Kyun Kim. Lecture 13-14: Face recognition, subspace/manifold learning. *EE462: Machine Learning for Computer Vision*, pages 5–16, 2014.

[25] Jonathon Shlens. A tutorial on principal component analysis. http://www.cs.cmu.edu/ elaw/papers/pca.pdf, 2005.

[26] Duncan Gillies. Principal component analysis. *Intelligent Data Analysis and Probabilistic Inference*, 15, 2015.

[27] GAIA Group of Artificial Intelligence Applications. jcolibri. http://gaia.fdi.ucm.es/research/colibri/jcolibri. Accessed on 18 Jul 2015.

[28] Yale University Department of Statistics. Multiple linear regression, statistics 101-103. http://www.stat.yale.edu/Courses/1997-98/, 1997.

[29] Tae-Kyun Kim. Lecture 9-10: Object recognition, categorisation / maximum margin classifier. *EE462: Machine Learning for Computer Vision*, pages 38–39, 2014.

[30] Andre Kushniruk, Vimla Patel, and David Fleiszer. Analysis of medical decision making: a cognitive perspective on medical informatics. In *Proceedings of the Annual Symposium on Computer Application in Medical Care*, page 193. American Medical Informatics Association, 1995.

[31] Derek Moore. Antibiotic classification mechanism. http://www.orthobullets.com/basic-science/9059/antibiotic-classification-and-mechanism, 2015.

[32] Christopher D Manning, Prabhakar Raghavan, Hinrich Schütze, et al. *Introduction to information retrieval*, volume 1. Cambridge university press Cambridge, 2008.

[33] SRS Pharmaceuticals Private Limited. Narrow spectrum antibiotics. http://www.srspharma.com/narrow-spectrum-antibiotics.htm, 1995 – 2010.

A

ATTRIBUTES AND THEIR IMPORTANCE

The following table shows the attributes considered and their relative importance is ranked. A higher rank of importance indicates that the attribute is more important, i.e. 'Organs Infected' is the most important, and 'Gender' is the least important.The table also shows the amount of documentation for each attribute in the case base. Note that only attributes that had at least 50% of data available were considered in the IDSS.

Attribute	Importance	Data Available?
Organs Infected	1	No
Bugs	2	Limited (18%)
Wound Cultures	2	Very Limited (3%)
Blood Cultures	2	Very Limited(4%)
Urine Cultures	2	Very Limited (2%)
CSF Cultures	2	No
Sputum	3	Very Limited (4%)
Allergies	4	Very Limited (7%)
HIV	5	Very Limited (1%)
Diabetic	6	Limited (11%)
Pregnant	7	No
Days in Hospital	8	No
Age	9	Yes
Renal Support	10	Very Limited (7%)
CREA	11	No
ALT	11	No
ALP	11	No
BILI	11	No

Table 14: Attributes and their relative importance, and the amount of data available in the case base

Attribute	Importance	Data Available?
Lactate	12	Most (67%)
CRP	12	No
WCC	12	No
Weight	13	No
Body Temperature	14	Limited (43%)
Ionotropes	14	No
Oxygen Requirements	14	Most (96%)
Steroids	14	No
Respiratory Rate	14	Most (97%)
Urinary Catheter	14	Yes
Central Line Changed	14	Very Limited (12%)
Ventilation Support	14	Yes
Abdominal Examination	14	Yes
Chest Examination	14	Yes
Chest Radiography	14	Yes
Gender	14	Yes

Table 15: Attributes and their relative importance, and the amount of data available in the case base

B

CLUSTERING OF DRUGS USING PCA

B.1 Standard Scaler

Attributes have been normalised using the standard scaler so that each attribute has a mean of 0 and variance of 1. PCA has been applied to reduce the dimensionality to 3, and the cases now projected into a 3-dimensional space. Figures 26 and 27 show this projection, and cases where a drug is prescribed are in green.

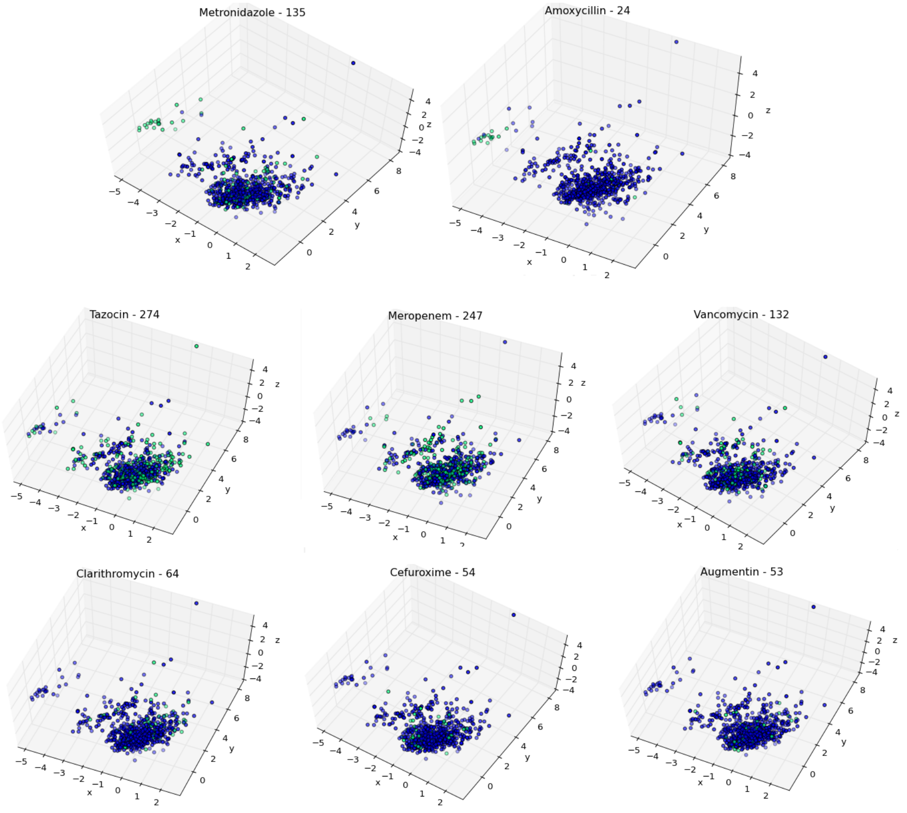
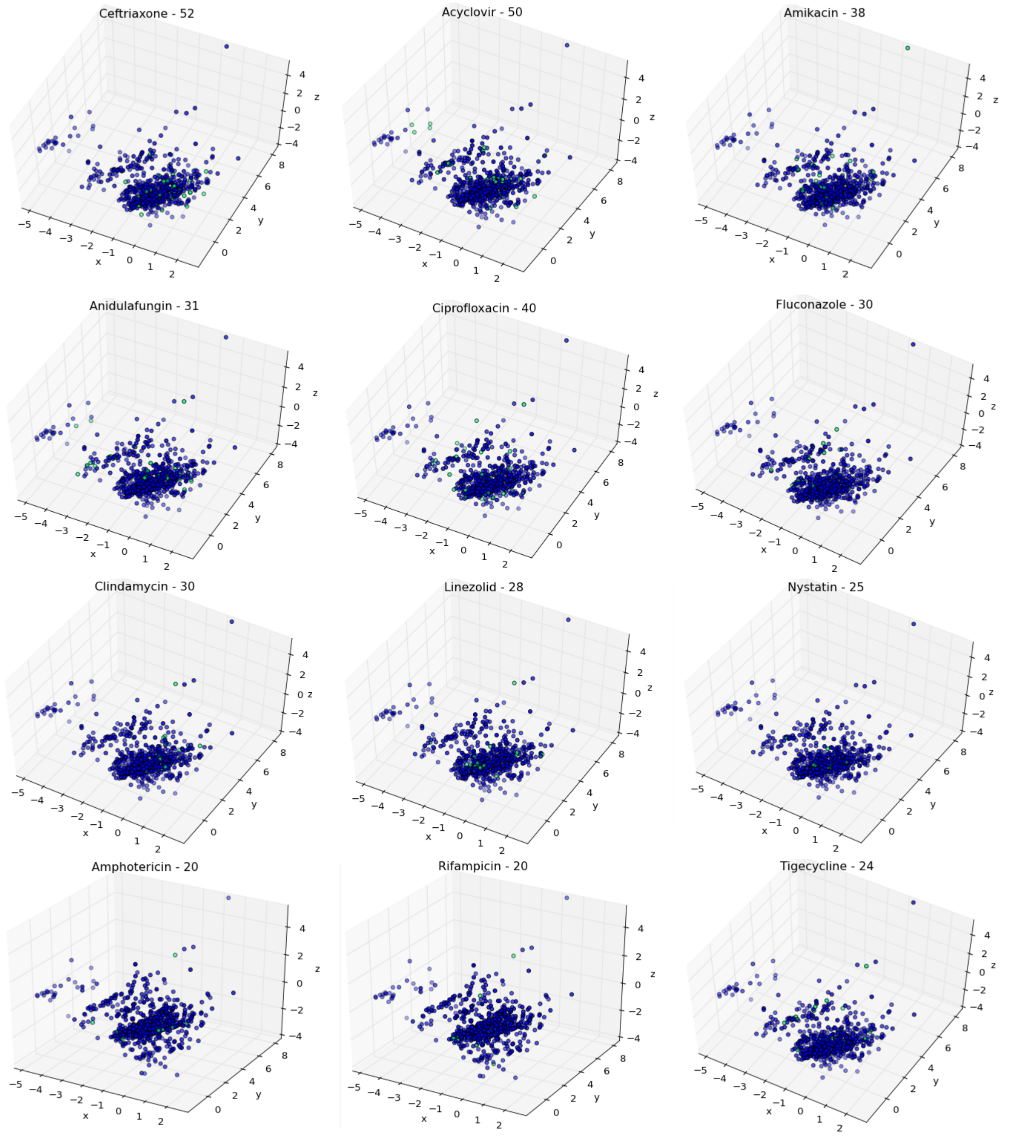


Figure 26: Clustering using PCA with standard scaler. Cases where drugs are prescribed are in green.


Ceftriaxone - 52

Acyclovir - 50

Amikacin - 38

Anidulafungin - 31

Ciprofloxacin - 40

Fluconazole - 30

Clindamycin - 30

Linezolid - 28

Nystatin - 25

Amphotericin - 20

Rifampicin - 20

Tigecycline - 24

Figure 27: [continued] Clustering using PCA with standard scaler. Cases where drugs are prescribed are in green.

B.2 Minmax Scaler

Attributes were normalised using the Minmax scaler so that each attribute has a value between 0 and 1. PCA was applied to reduce the dimensionality to 3. Figures 28 and 29 show the projection of the cases into a 3-dimensional space, and cases where a drug is prescribed are in green.

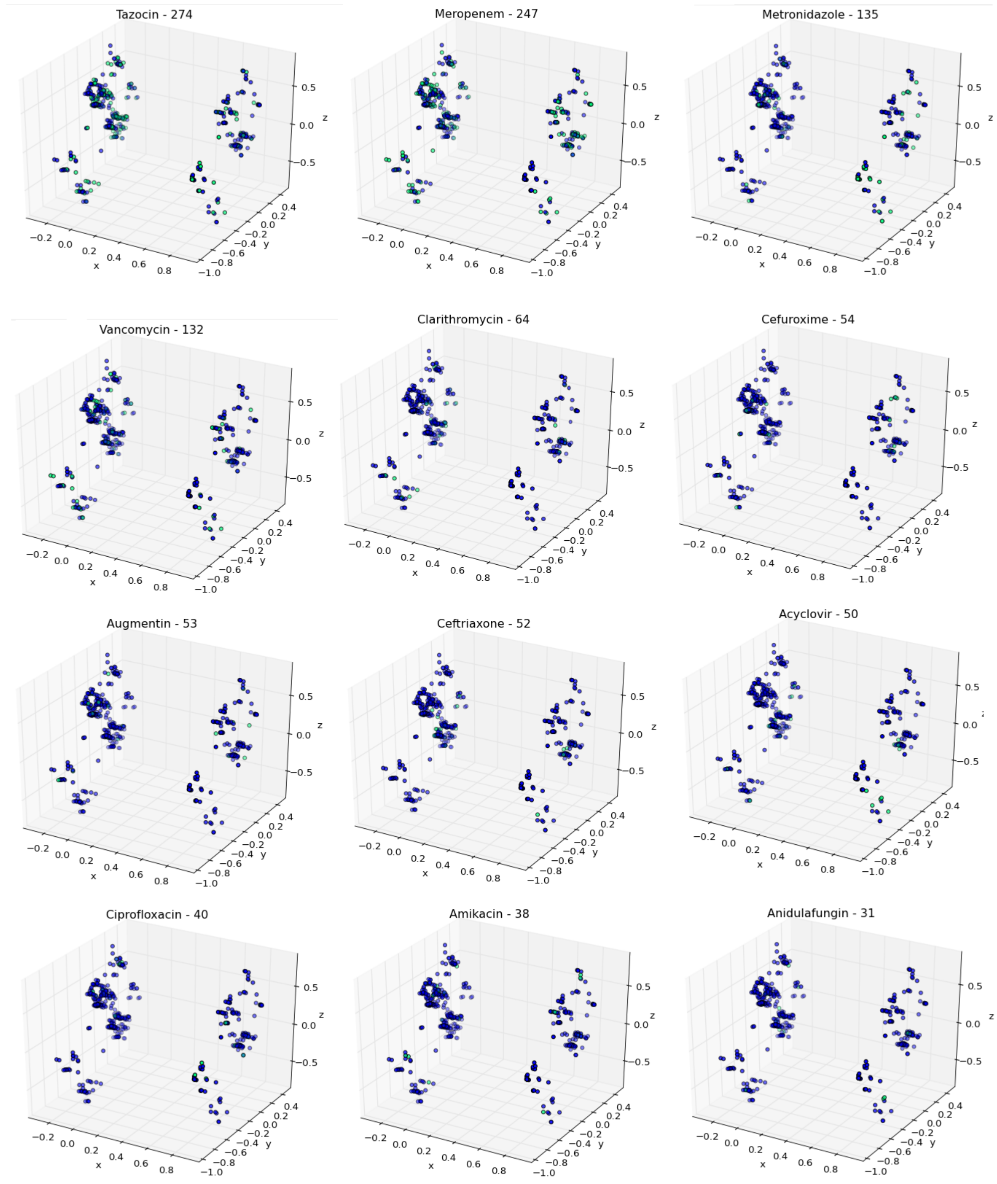


Figure 28: Clustering using PCA with Minmax scaler. Cases where drugs are prescribed are in green.

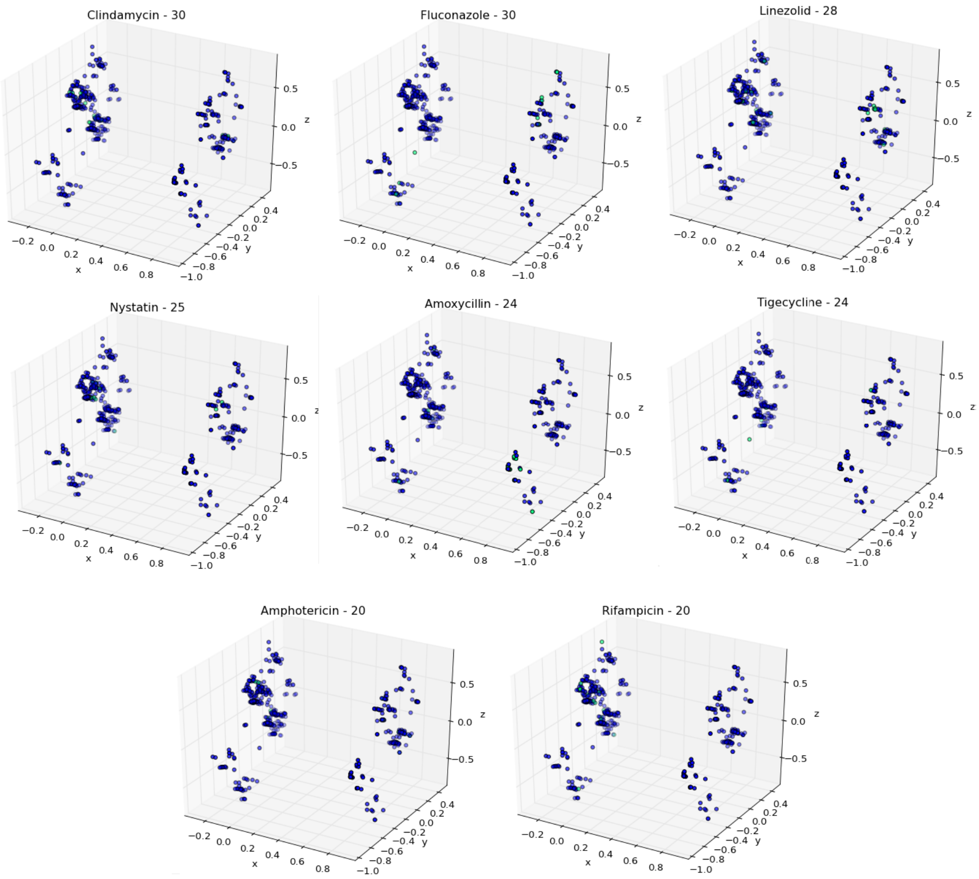


Figure 29: [continued] Clustering using PCA with Minmax scaler. Cases where drugs are prescribed are in green.