Master in Intelligent Interactive Systems
Universitat Pompeu Fabra

Predicting Multi-Resistance of Bacteria in an Intensive Care Unit

Àlvar Hernàndez Carnerero

Supervisor: Miquel Sànchez i Marrè

Co-Supervisor: Vicenç Gómez

July 2020

Contents

1 Introduction **1**

1.1 Overview 3

2 Background **5**

2.1 Feature weighting techniques 6

2.2 Temporal management 9

2.3 Class imbalance distribution 10

3 Methods **13**

3.1 Data description 13

3.2 Generation of new features 16

3.2.1 Past cultures features 16

3.2.2 ICU bacteria features 18

3.3 Data preprocessing 19

3.4 Experimental work 21

4 Results **29**

4.1 Feature weighting 29

4.2 Incremental training window evaluation 37

4.3 Generated features evaluation 39

4.4 Feature Selection 43

4.5 Oversampling and feature selection 46

4.6 Temporal evolution of prediction 46

4.7 MDR predictor 47

5 Discussion **53**

5.1 Future work 55

List of Figures **56**

List of Tables **57**

Bibliography **58**

Dedication

I would like to dedicate this work to my family for supporting me, to my partner for believing in me and for his medical advice, to my supervisor for being my mentor since before ending my degree and for teaching me many of the things I know about machine learning, and my co-supervisor for his guidance.

Acknowledgement

We are thankful to Joaquín Álvarez-Rodríguez from University Hospital of Fuenlabrada, Madrid, Spain for providing the database used in this research as well as for his support and to Inmaculada Mora-Jiménez, Cristina Soguero-Ruiz and Sergio Martínez-Agüero from the DASES research group from the University Rey Juan Carlos (URJC), for their joint collaboration in this research work. This work has been partly supported by the Spanish Thematic Network "Learning Machines for Singular Problems and Applications (MAPAS)" (TIN2017-90567-REDT, MINECO/FEDER EU).

Abstract

This study considers the prediction of "multi-drug" resistance (MDR) of Pseudomonas aeruginosa bacterium caused by nosocomial infections in the Intensive Care Unit (ICU). An ensemble of binary classifiers implemented with different Machine Learning (ML) methods is applied for prediction using as training data health records and past sensitivity tests (antibiogram) results. This work proposes to generate two new types of features to improve predictor's performance. The first one is based on using information of previous antibiograms of a particular patient to predict their future resistance to antibiotics. The second kind of features employs bacterial information from the rest of the patients in the ICU to predict the antimicrobial resistance for a certain patient. In addition, in the study it is suggested to use a training window with incremental size so that training set is always temporarily as near as possible to the test instances to be predicted. Some techniques such as feature selection and oversampling are also used to further improve efficiency and accuracy. Results show that using an incremental window for training improves success rates in the domain of this problem, and expose that knowing the outcomes of past antibiograms, substantially improves prediction. It is also observed that considering resistant bacteria present in the ICU is useful to anticipate antimicrobial resistance. From these results it is further inferred that resistant bacteria may be spreading among patients in the ICU within populations that rapidly mutate, which can induce non-stationary in the data distribution. It is concluded that using these contributions, experiments show promising results in MDR prediction even using simple features and limited training data.

Keywords: Temporal modelling; Machine learning; Classification problem; Clinical data; Antibiogram; Antimicrobial resistance; Intensive care unit; Health care

Chapter 1

Introduction

Antimicrobial resistance has been increasing for decades, and the rate at which new antibiotics are synthesized is not as fast as it would be required to prevent this trend [1, 2]. A large proportion of infections caused by resistant bacteria occurs during hospital stays, specially in the Intensive Care Unit (ICU) [3], where infection rates are much higher than in other hospital divisions [4]. This is due to its severely vulnerable population and to the high risk of becoming infected through multiple procedures and the use of invasive devices distorting the anatomical integrity-protective barriers of patients (intubation, mechanical ventilation, vascular access, etc.) [5]. Infections produced during the admission of a patient in the hospital are referred as *nosocomial infections*. It is generally considered that nosocomial infections arise 48 hours after admission [6].

In the ICU it is frequent to find some kinds of bacteria which can become multidrug-resistant. The most common are *Acinetobacter spp.*, *Enterococcous fecalis* and *Enterococcus faecium*, *Escherichia coli*, *Klebsiella pneumoniae*, *Pseudomonas aeruginosa* and *Staphylococcus aureus*. This work is focused on *Pseudomonas aeruginosa* due to their prevalence and virulence, being a bacterium commonly responsible for nosocomial infections. It is naturally resistant to many antibiotics and has a remarkable capacity for acquiring new resistance mechanisms, creating therapeutic problems [7]. *Pseudomonas aeruginosa* is considered to be "multi-drug" resistant (MDR) when it is

observed a reduced in vitro susceptibility to three or more antimicrobial families [8].

Infections due to MDR microorganisms are a major problem. They have a significant impact in the ICU, where they cause additional morbidity, mortality, and health care costs [5, 9]. Particularly, inappropriate initial antimicrobial treatment of *Pseudomonas aeruginosa* infection is associated with statistically greater mortality compared to initial treatment with an antimicrobial regimen to which the bacterium is susceptible. In turn, the growing MDR rate of *Pseudomonas aeruginosa*, increases the chance of an inappropriate initial antimicrobial treatment [10].

To treat nosocomial infections in the hospital, at first a culture or microbiological analysis is normally performed. In this investigation germs are isolated and, for each of them, an antibiogram is built. An *antibiogram* represents the in vitro bacterium's resistance to a series of antibiotics. The set of antibiotics used in an antibiogram can be selected for the specific type of bacterium being tested. The result of the test is presented as a vector of couples (antibiotic/sensitivity) [11]. Antibiograms are often used by clinicians to assess bacteria susceptibility rates, as an aid in selecting empiric antibiotic therapy [12], and avoid an incorrect treatment. Hence, an antibiogram would vary within a given bacterium species, depending on the different types of resistance a particular bacterium has developed to different antibiotics. However, quite often groups of antibiotics still have similar sensitivity when tested on a given bacterium species [13].

The result of the antibiogram is also helpful to decide whether it is needed to isolate a specific patient so their bacteria does not spread among the rest of ICU population. It has been observed that one of the most relevant factors of the spread of bacterial resistance is the so called cross-transmission which may facilitate the spread of resistant bacteria from one patient to another. Some of the ways to prevent cross-transmission to happen are, for instance, appropriate hand hygiene, skin cleansing, and contact precautions [14], and even more important is information about bacteria in the ICU and their resistance is key to know how and where to extreme caution. The result of an antibiogram can take from 24h to 48h to be obtained [15]. In the ICU environment this time is crucial specially for patients with critically adverse

health conditions, and high risk of mortality. Having this result in time, may make the difference not only in saving patient's lives but also preventing resistant bacteria to spread.

Because of the aforementioned reasons, in the current study it is proposed to use Machine Learning (ML) techniques, and special temporal data treatment to get a quick approximation of the antibiogram result.

1.1 Overview

The document is arranged as follows. In the current chapter the main concepts related to the problem being tackled are described. In chapter 2 it is discussed the motivation of this work as well as previous studies in the field. Chapter 3 begins with an explanation of the data and features that are going to be used, together with the generation of new features. Later, the preprocessing applied to the data is detailed, and finally it is explained the set of performed experiments and how they are carried out. In chapter 4 the results of the experiments are expressed and analyzed. Chapter 5 contains the discussion which deepens into the meaning, importance and relevance of the results. It includes the overall conclusion and exposes the future work.

Chapter 2

Background

This study arises from the collaboration of the Knowledge Engineering and Machine Learning Group (KEMLG-UPC) from the Universitat Politècnica de Catalunya with the DASES research group from the Universidad Rey Juan Carlos (URJC) and the University Hospital of Fuenlabrada, Madrid, Spain. This collaboration has the purpose of predicting "multi-drug" resistance to antibiotics of bacteria present in the ICU. The resulting studies have led to the elaboration of a paper [16] accepted at the Singular Problems for Health Care (SP4HC) Workshop at the 24th European Conference on Artificial Intelligence (ECAI 2020).

Many of the state-of-the-art studies regarding the use of ML methods to predict antimicrobial susceptibility use whole genome sequencing [17, 18, 19, 20]. Despite it is a very promising technique, it involves very significant costs. As an alternative, this work uses the information from the ICU health records and demographic data of patients, along with historic antibiogram results to train a ML model, aiming to predict resistant bacteria in new cultures. As opposed to the whole genome sequencing, the current approach intends to use data which is already available in the vast majority of hospitals, in order to speed up the process of identifying MDR cases. Similar strategies have been analyzed in the past [15, 13, 21].

For the application of ML methods the use of some techniques was necessary, such as feature weighting techniques, temporal management techniques and class imbalance

distribution approach. These techniques are described in the next sections.

2.1 Feature weighting techniques

Feature weighting is used in this study as a first approach to evaluate the quality and relative relevance of variables that are going to be used as indicators of antimicrobial resistance. Three feature weighting methods are selected seeking good performance and taking into account different mechanisms to calculate feature relevance. The chosen methods from literature fulfilling these requirements are *Information Gain* (IG) [22], *Class-Value Distribution* (CVD) [23] and *Unsupervised Entropy-Based method 1* (UEB-1) [24].

The **IG** algorithm aims to assign lower weights to features containing less information of the target to be predicted and higher to features with greater amount of information. This metric is designed to evaluate each feature by its ability to reduce the overall information entropy of the data set. The data set information entropy is computed by the formula in Equation 2.1 where p_i (the probability of class i) is estimated by its relative frequency in the data set D:

$$E(D) = -\sum_{i} p_i \log_2 p_i. \tag{2.1}$$

Now, for each feature it is calculated what is the information gain of knowing its value. To do this the average information entropy for the feature is computed, and it is subtracted from the just calculated data set information entropy. The average information entropy for a feature is computed by averaging the information entropy of the data set restricted to each possible value for the feature, as follows:

$$E(D_{[f]}) = -\sum_{v_i \epsilon V} E(D_{[f=v_i]}) \frac{|D_{[f=v_i]}|}{|D|}, \tag{2.2}$$

in Equation 2.2, the expression $E(D_{[f=v]})$ refers to instances in the data set that

have value v for feature f. Then, information gain is obtained by Equation 2.3:

$$G(f) = E(D) - E(D_{[f]}). \tag{2.3}$$

The values obtained for each feature can be used as a global weight with an appropriate scaling process.

The **CVD** algorithm belongs to the correlation-based global weighting algorithms, which are based on assigning higher weights to features showing greater correlation between the value distribution and the class distribution in the data set. The process to calculate the array of weights is the following:

The first step to calculate the array of weights is to fill a correlation matrix for each feature, representing the correlation between feature's values and class values. In this matrix, rows are the possible values the feature can take, and columns are the classes to which an instance may belong. Each value q_{ij} in the correlation matrix represents the number of instances that have value V_i for the particular feature, and belong to class C_j.

CVD metric considers two aspects, the distribution of the feature's values across the classes, which is represented by a row, and the values associated to a class, represented by a single column. The perfect feature can be seen as a diagonal matrix, in which each row as well as each column has just a single value different to zero. In this case, a single feature's value would predict a specific class, and a class is determined by a particular feature. To take into account both class and value distribution Equation 2.4 is used.

$$H_a = \frac{1}{n} \sum_{i=1}^{n} \left(\frac{q_{\max,i}}{q_{+,i}} * \frac{q_{\max,i}}{q_{\max,+}} \right), \tag{2.4}$$

where n is the number of classes, $q_{+,i}$ is the number of instances belonging to class i, $q_{\max,i}$ is the maximum value of the column of class i and $q_{\max,+}$ is the number of instances that have the maximum value of $q_{\max,i}$.

Finally, the weight of the feature is obtained as in Equation 2.5:

$$W_a = \frac{H_a - \frac{1}{|a|*n}}{1 - \frac{1}{|a|*n}}, \tag{2.5}$$

where $|a|$ is the number of different feature values and n is the number of classes.

UEB-1 is an unsupervised feature weighting method based in entropy computations. It relies on the observation that removing an irrelevant feature from the feature set may not change the underlying concept of the data, but it should change it otherwise. The first step consists in computing the entropy for the entire data set of N instances, which is given as Equation 2.6:

$$E = -\sum_{i=1}^{N}\sum_{j=1}^{N} S_{ij} * \log_2(S_{ij}) + (1 - S_{ij}) * \log_2(1 - S_{ij}), \tag{2.6}$$

where S_{ij} is is the similarity value between the instance i and the instance j normalized to $[0, 1]$. In this study, where all features are numeric, the similarity between two instances is $S_{ij} = e^{-\alpha * D_{ij}}$, where D_{ij} is the Euclidean distance between instances i and j, and α is computed as $\alpha = \frac{-\ln 0.5}{\bar{D}}$, where $\bar{D}$ is the average distance among all the instances.

For a dataset of M features, the algorithm computes the entropy of the data M times, each time removing one different feature. These M entropy values are stored in an array P, and each entropy value is accessed as $P_i, i = 1 \ldots M$. Finally, the algorithm takes the entropy value computed for each feature, and applies a scaling process to assign the weights. In Equation 2.7 scaling is carried out, returning weights in $[0, 1]$ range, for each feature i.

$$w_i = \frac{P_i - \arg\min(P)}{\arg\max(P) - \arg\min(P)}. \tag{2.7}$$

Following this process, features are ranked in descending order of relevance.

2.2 Temporal management

Antimicrobial resistance is a phenomenon that changes over time as bacteria mutates. It allows bacteria to be more resistant to antibiotics as time passes by. As previously mentioned, the variables considered include health records from ICU patients, demographic data and antibiogram results which is what has to be predicted. Since bacteria's mutations are not amongst the available features, as time passes by, the feature's values telling apart one class from another may change. For example, if two considered instances have the exact same values for all of features at disposal, but they belong to different moments in time, their classification may be different depending on how the resistance has evolved during this time.

This fact has been previously described as the *concept drift* in which the concept of interest may depend on some hidden context, not given explicitly in the form of predictive features. Changes in the hidden context over time can induce more or less radical changes in the target concept. Changes in hidden context may not only result in a change of the target concept, but may also cause a change of the underlying data distribution [25].

In [13] it is proposed to use *instance selection* to handle concept drift as it is the most commonly used technique and has been found to offer good results. More specifically, it is proposed to use a method based on instance selection that consists in generalizing from a window that moves over recently arrived instances and uses the learnt concepts for prediction only in the immediate future. This method is called *windowing*.

Nevertheless, the authors of [13] also stated that when using windowing *local concept drifts* may appear. *Local concept drift* are changes in the concept or data distribution that occur in some regions of instance space only, with types and severity that may depend on the location in the instance space. To deal with *local concept drift*, *dynamic selection* (DS) techniques are suggested.

DS consists in selecting one or more base classifiers for each query instance to be

classified. Base classifiers are selected from a pool of classifiers by estimating their competence level. Only the most competent, or an ensemble containing the most competent classifiers, is selected to predict the label of a specific test sample. These type of techniques are based on the notion that not every classifier in the pool is an expert in classifying all unknown samples, it assumes that each base classifier is an expert in a different local region of the feature space [26].

DS methods can be divided in two types of algorithms: *Dynamic Classifier Selection* (DCS), in which just one base classifier (the one with highest competence level) is selected for the query classification, and Dynamic Ensemble Selection (DES), in which an ensemble of base classifiers are selected to predict query instance. This study uses a DES approach, particularly K-nearest-oracles (KNORA), since it has been shown to perform a little better than the other DCS schemes [27]. For any test instance, KNORA finds its nearest neighbors in the validation set. After that, it figures out which classifiers correctly classify those neighbors in the validation set and uses them as the ensemble for classifying the given pattern in that test set.

2.3 Class imbalance distribution

The data set used in the current study suffers from class imbalance, that is, there is not the same number of sensitive and resistance instances. This problem is even more evident when using windowing, causing, in many cases, the minority class contain too few instances for a model to effectively learn the decision boundary. To overcome this problem, one of the proposed solutions is oversampling. To generate new instances belonging to the minority class, the *Synthetic Minority Oversampling TEchnique* (SMOTE) [28] is used.

SMOTE works as follows. The algorithm first selects, at random, an instance from the minority class. After that, instances belonging to the k nearest neighbors of this particular instance are found (typically $k = 5$). One of the neighbors is randomly chosen, and a synthetic example is created at a randomly selected point in a line connecting the two instances in feature space.

More specifically, an extension of the SMOTE algorithm is used, which is Borderline-SMOTE [29]. This extension is chosen because it reports the best results for this particular problem. It consists in selecting those instances of the minority class that are misclassified. Then, it oversamples just those instances hard to classify, providing more resolution only where it may be required. These misclassified instances tend to be ambiguous and located near or on the border of the decision boundary, where class membership may overlap.

Chapter 3

Methods

This chapter details data, features and experiments employed in the study. In Section 3.1, the data set used is described together with the selected features. Section 3.2 indicates how new features are generated using temporal information contained in the data set. The preprocessing applied to the data is specified in Section 3.3. Finally, in Section 3.4 the experiments carried out are discussed.

3.1 Data description

Data considered in this work is a unified and anonymized dataset specifically collected for the study of antimicrobial resistance in the ICU of the University Hospital of Fuenlabrada (UHF) in Spain. The data set covers years from 2004 to 2016. During this time interval, $2,617$ patients were admitted to the ICU, and $32,787$ cultures were carried out from $3,016$ admissions. It has a number of 257 different types of germs and 26 antimicrobial families.

The data set contains the results of antibiograms carried out to patients in the ICU, that is, the results of the sensitivity tests (sensitive s or resistant r) for a certain germ and antibiotic used in the test. It also includes demographic data of the patients and information about their ICU admission.

As already mentioned, this study is focused on just one type of bacteria among

the multiple available in the data set: *Pseudomonas aeruginosa*. This bacterium is considered MDR if it is resistant to three or more of the following antimicrobial families within the same culture: *Aminoglycosides* (AMG), *Carbapenems* (CAR), *4th Generation Cephalosporins* (CF4), *Extended-spectrum penicillins* (PAP), *Polymyxins* (POL) and *Quinolones* (QUI).

In this work, all instances containing the bacterium and antimicrobial families of interest are analyzed. Then, each instance is represented by a set of selected features described in Table 1. Features `c&amg`, `c&car`, `c&cf4`, `c&pap`, `c&pol` and `c&qui` represent the targets to be predicted, this is, the result of the sensitivity test for *Pseudomonas aeruginosa* to the antimicrobial families of AMG, CAR, CF4, PAP, POL and QUI, respectively.

Table 1: Features names and their descriptions.

Feature name	Description
c&amg	Result of the test to the AMG family (r/s).
c&car	Result of the test to the CAR family (r/s).
c&cf4	Result of the test to the CF4 family (r/s).
c&pap	Result of the test to the PAP family (r/s).
c&pol	Result of the test to the POL family (r/s).
c&qui	Result of the test to the QUI family (r/s).
days_to_culture	# days elapsed from admission to the date of the culture.
date_culture	Date of the culture.
number_antibiotics	# antibiotics tested in the antibiogram.
culture_type	Type of culture performed (pharynx, urine, blood, etc.).
culture_type_grouped	Type of culture performed grouped (respiratory, urine, surface, etc.).
culture_type_grouped_2	Type of culture performed grouped (clinical sample/surface).

Continuation of Table 1

Feature name	Description
day_week_culture	Day of the week when culture is carried out.
day_month_culture	Day of the month when culture is carried out.
month_culture	Month on which culture is carried out.
year_culture	Year on which culture is carried out.
origin	Clinical origin before ICU admission.
reason_admission	Reason of admission at ICU.
goi_A	Group of illness A. Cardiovascular events.
goi_B	Group of illness B. Kidney failure, arthritis.
goi_C	Group of illness C. Respiratory problems.
goi_D	Group of illness D. Pancreatitis, endocrine.
goi_E	Group of illness E. Epilepsy, dementia.
goi_F	Group of illness F. Diabetes, arteriosclerosis.
goi_G	Group of illness G. Neoplasms.
pluripathology	# of groups of illness to which patient belongs.
patient_category	Patient's clinical category.
age	Patient's age.
gender	Patient's gender.
start_date	Date when patient's admission begins.
day_week_admission	Day of the week when patient's admission begins.
day_month_admission	Day of the month when patient's admission begins.
month_admission	Month when patient's admission begins.
year_admission	Year when patient's admission begins.

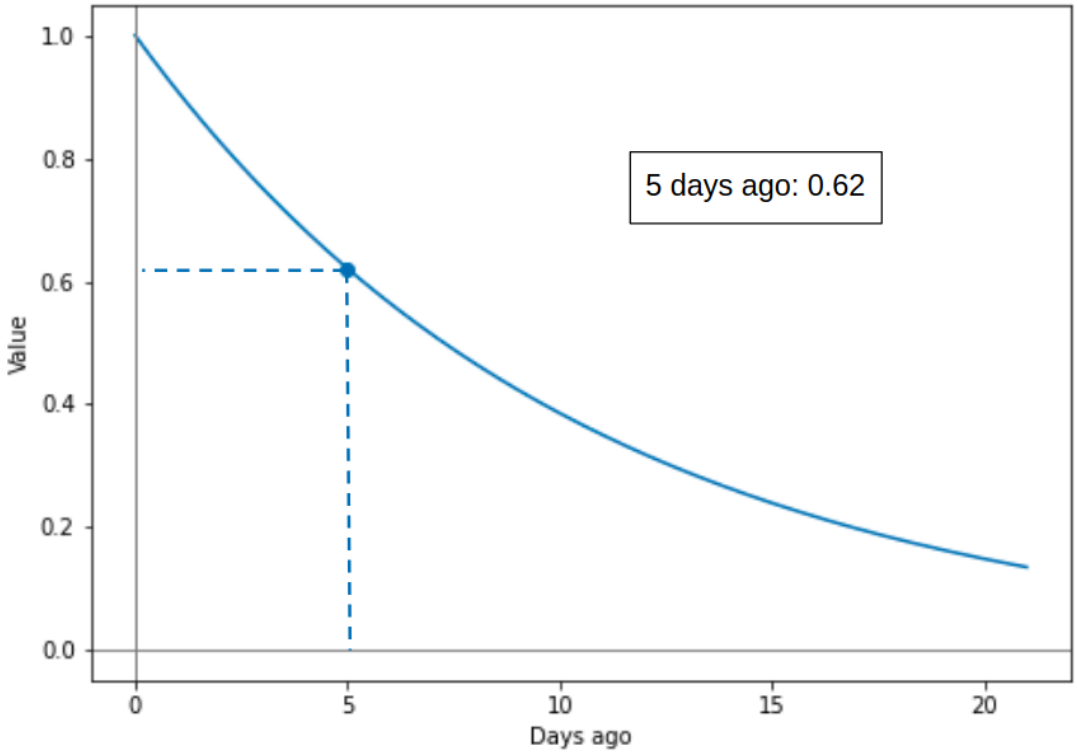
3.2 Generation of new features

In addition to the selected features, the current study proposes to generate new features based on the temporal information contained among cultures in the data set. The new features are divided in two types: the ones representing results of patient's past antibiograms, and the ones expressing resistance of bacteria present in the ICU.

3.2.1 Past cultures features

These features indicate the detection of resistant bacteria in previous antibiograms for a specific patient, and reflect the time passed since detected. For a particular feature, a single value is calculated by aggregating the result from past antibiograms of *Pseudomonas aeruginosa* during a specific time interval. Their value should decay quickly in time. Therefore, an exponential decay of the form of Fig. 1 is used in their calculation.

Figure 1: Exponential decay



Six features of this kind are generated. Each feature f_i will take into account one of the aforementioned antimicrobial families a_i, where $i\epsilon[1..6]$. These features are `p&amg`, `p&car`, `p&cf4`, `p&pap`, `p&pol` and `p&qui`. For instance, in the case of `p&amg`, only the cultures containing the result of the sensitivity test to the AMG family will be selected.

For a specific instance of the data set representing a culture c of patient p, it is collected the set of cultures containing *Pseudomonas aeruginosa* that have been carried out to patient p during their admission in the ICU, between 21 days and 48 hours before the date d of the culture c.

Note that, since the results of the test usually take 48 hours to be provided, it is not possible to use cultures taken, for instance, one hour ago. Apart from that, from a clinical viewpoint, if the culture result is positive, it is kept as positive for the next 21 days. For this reason, cultures collected 21 days before the date d of the current culture c are considered.

The set of past cultures collected is splitted into six subsets. Each subset C_i contains cultures tested just for a particular antimicrobial family a_i and it is the set used to calculate the value of its respective feature f_i. C_i contains n cultures, and each culture $c_j\epsilon C_i$ has a sensitivity test result r_j, which is 0 or 1 depending on whether bacterium is sensitive or resistant to a_i respectively, and a date in which it was performed d_j, where $j\epsilon[1..n]$. To weight the contribution of a culture c_j the negative exponential distribution (ED) is employed as follows:

$$ED_{\{c_j,c\}} = \begin{cases} 0 & \text{if } r_j = 0 \\ n^{-(d-d_j)} & \text{if } r_j = 1 \end{cases}, \tag{3.1}$$

where n is a real number experimentally set to 1.1. Then, to compute the value of each of the six features f_i for the instance's culture c linked to patient p, the

maximum outcome in Equation 3.1 is determined according to Equation. 3.2.

$$f_{i\{C_i,c\}} = \max_{c_j \epsilon C_i} ED_{\{c_j,c\}} \tag{3.2}$$

3.2.2 ICU bacteria features

The purpose of this second type of features is to capture the presence of resistant bacteria in the ICU and their *intensity*. By *intensity*, it is considered how long ago the resistant bacterium was detected on patients and the number of patients in which it was detected. It is calculated by considering the result of past sensitivity tests of *Pseudomonas aeruginosa* for the patients in the ICU, during a particular time interval. Again, the computation of these features is based on an exponential distribution.

A total of six features of this kind are built, as before, one f_i for each antimicrobial family a_i. These features are `r&amg`, `r&car`, `r&cf4`, `r&pap`, `r&pol` and `r&qui`.

Considering an instance containing a culture c that belongs to a particular patient p, the time interval considered to gather the set of cultures is, again, between 21 days and 48 hours before the date of culture c but now the set of cultures considered is extracted from those carried out to all n patients in the ICU except for patient p. Each patient in the ICU is denoted as p_j where $j\epsilon[1..n]$.

The group of past cultures is divided into six subsets, as before, one subset C_i for each feature f_i where $j\epsilon[1..6]$. In this case, every particular subset C_i, is splitted into n smaller sets of cultures C_j, in which all the cultures belong to the same patient p_j. The set of cultures of the patient under analysis p are excluded as previously mentioned. Each C_j is conformed by m cultures expressed as c_k where $k\epsilon[1..m]$. Every c_k has result r_k and date d_k. The formula in which the negative exponential distribution is applied is equivalent to Equation 3.1, just replacing c_j, d_j and r_j by c_k, d_k and r_k respectively. Each feature f_i is calculated by adding up the maximum

value in Equation 3.1, for each patient p_j as showed in Equation 3.3.

$$f_{i\{C_i,c\}} = \sum_{C_j \epsilon C_i} \max_{c_k \epsilon C_j} ED_{\{c_k,c\}} \tag{3.3}$$

3.3 Data preprocessing

To proceed with the model design for the six different targets (features `c&*`), six data sets are created, one associated to each target. These data sets will be used to train a ML model for each label to be predicted. Training six different classifiers instead of, for instance a multi-class classifier, allows each classifier to be specialized in predicting its particular target, therefore tuning classifier's hyperparameters individually. In order to limit the study to MDR adquired during admission in the ICU (as a nosocomial infection), only instances of patients admitted in the ICU for more than 48h are considered.

The final data set for `c&amg` is composed by 696 cultures, the data set for `c&car` contains 582, the data set for `c&cf4` contains 688, the data set for `c&pap` contains 690, the data set for `c&pol` contains 426 and the data set for `c&qui` contains 690 cultures. All six data sets have 41 features including the target one, and each has 12 features with missing values. These 12 features are the proposed features, as previously seen, there are six for each of the two types. The percentages of *missing values* are depicted in the Tables 2 and 3 for `p&*` and `r&*` features respectively.

Table 2: Missing values proportion in p&* features.

Data set family	p&amg	p&car	p&cf4	p&pap	p&pol	p&qui
AMG	33	41	33	33	51	34
CAR	31	35	31	31	49	32
CF4	34	41	34	34	52	34
PAP	34	41	34	34	52	34
POL	27	32	27	27	32	27
QUI	34	41	34	34	52	34

Before training the models, missing data is addressed. In the case of `p&*`, note that

Table 3: Missing values proportion in r&* features.

Data set family	r&amg	r&car	r&cf4	r&pap	r&pol	r&qui
AMG	13	19	13	13	33	14
CAR	14	18	14	14	36	15
CF4	13	19	13	13	34	14
PAP	13	19	13	13	33	14
POL	12	17	12	12	19	13
QUI	13	19	13	13	34	14

the patient may not have any previous culture for *Pseudomonas aeruginosa* and a particular antimicrobial family for some time intervals, and therefore, a missing value is considered for that feature. This fact can be addressed following different approaches, such as deleting instances with missing data or imputing missing values [30]. In this study it is proposed an strategy based on the clinical meaning of the generated features. The smaller the value of these features, the more time will have passed since a particular patient was infected. As a result, if `p&*` features do not have any value, it suggests that no *Pseudomonas aeruginosa* has been recently detected for the patient. Therefore, very likely the patient would not have been recently infected with a resistant bacterium, and the value provided by Equation (3.2) should be very small. In this case, missing values are replaced by a 0.

Similarly, regarding `r&*` features, a missing value is issued when there is no culture in the ICU as a whole for *Pseudomonas aeruginosa* and a particular antimicrobial family during a specific time interval. In this case, the smaller the value of `r&*` features, the fewer patients will have been infected with resistant *Pseudomonas aeruginosa* bacterium and the greater the elapsed time since they were infected. Therefore, if `r&*` features contain a missing value, it suggests that no *Pseudomonas aeruginosa* has been detected in the ICU, and very likely no patients would have been recently infected with a resistant bacterium. Hence, the value provided by Equation (3.3) should be very small. Again, in this case missing values are replaced by a 0.

Afterwards, all *categorical features are converted to binary* following a one-hot en-

coding strategy, except for the features representing dates. Since dates have an intrinsic ordering, smaller numerical values are assigned to further dates in the past, and greater values correspond to more recent dates.

Finally, *Pearson correlation* between features (without considering the targets) is calculated in order to discard the most correlated features, since they provide very similar information. The methodology is as follows. When a set of features have a correlation coefficient higher than 0.9 or lower than -0.9, just one of them is selected, discarding the rest. A visual representation of Pearson correlation between features for AMG data set is shown in Fig. 2. A similar correlation pattern between features is observed for the rest of the six data sets. For each data set, there is just one group of features that are correlated higher than 0.9, which are the following: `date_culture`, `year_culture`, `start_date` and `year_admission`. In this case, `date_culture` is selected because it is the most representative among them and the rest are discarded. In the end, all data sets have 38 features after this discarding including the target feature.

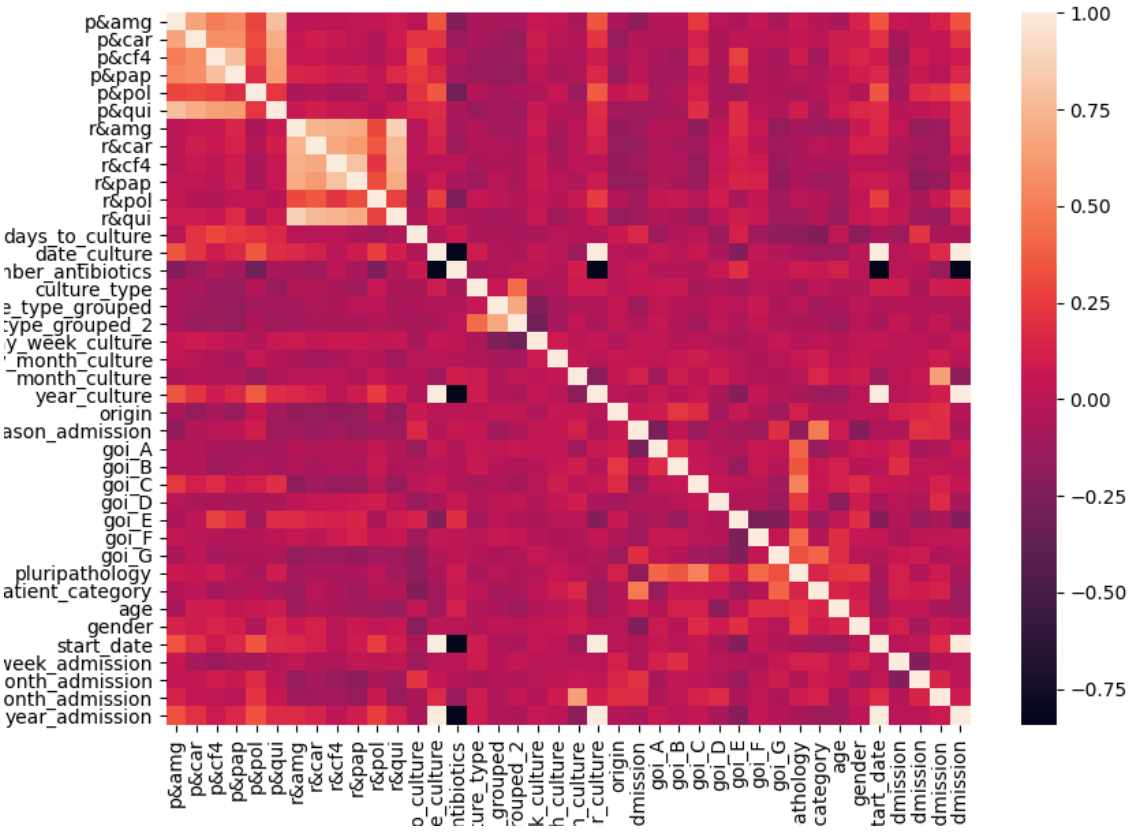
3.4 Experimental work

The first type of experiments, once the data is processed, evaluate the relevance of the set of features, both selected and generated, regarding the six different target features to be predicted. As a result, an array of weights is calculated for every target, with each weight of the array corresponding to particular feature of the data set. The weights are calculated by averaging the values returned by three different methods of feature weighting. The methods used are the ones discussed in Chapter 2: IG, CVD and UEB-1.

Looking now at the prediction of the target, the type of the problem proposed have associated a series of special characteristics that have to be considered in order to properly address it.

The first property is that health records have an inherent *temporal ordering*. This forces to use as training only instances that belong to a time prior to the test

Figure 2: Features correlation for `c&amg`



antibiograms that are to be predicted. In addition, a margin of time has to be respected between train and test windows, since results of antibiograms are not immediately available after they are carried out. As before, in this particular case, a time margin of 48h needs to be considered.

The second particularity encountered when predicting is the *concept drift*. The windowing technique described in [13], previously commented in Chapter 2, represents a very good approach to apply to the resolution of the problem analyzed in this study except for the third particularity of the data set.

The data scarcity makes it difficult to learn from temporal windows containing several months, even years. As previously mentioned, the number of cultures in the data sets ranges from 426 in the case of `c&pol` to 696 in `c&amg`. Taking into account those data sets represent cultures from 13 years (from 2004 to 2016), the average of cultures per year goes from 33 to 54 respectively, which is a relatively small number of instances considering how fast ICU bacteria are able to mutate and

change its sensitivity patterns. For this reason, it is proposed to use an *incremental window for training*, which increases its size as test window moves towards more recent instances. That is, the training window will be fixed from the first temporal instances, which are the oldest, and it will gradually increment in size, containing more instances, as more recent instances of test are predicted. The *test window*, on the other hand, will have a *fixed size* and it will progressively slide to select more recent instances.

The *incremental window scheme* requires training and test windows to have following characteristics:

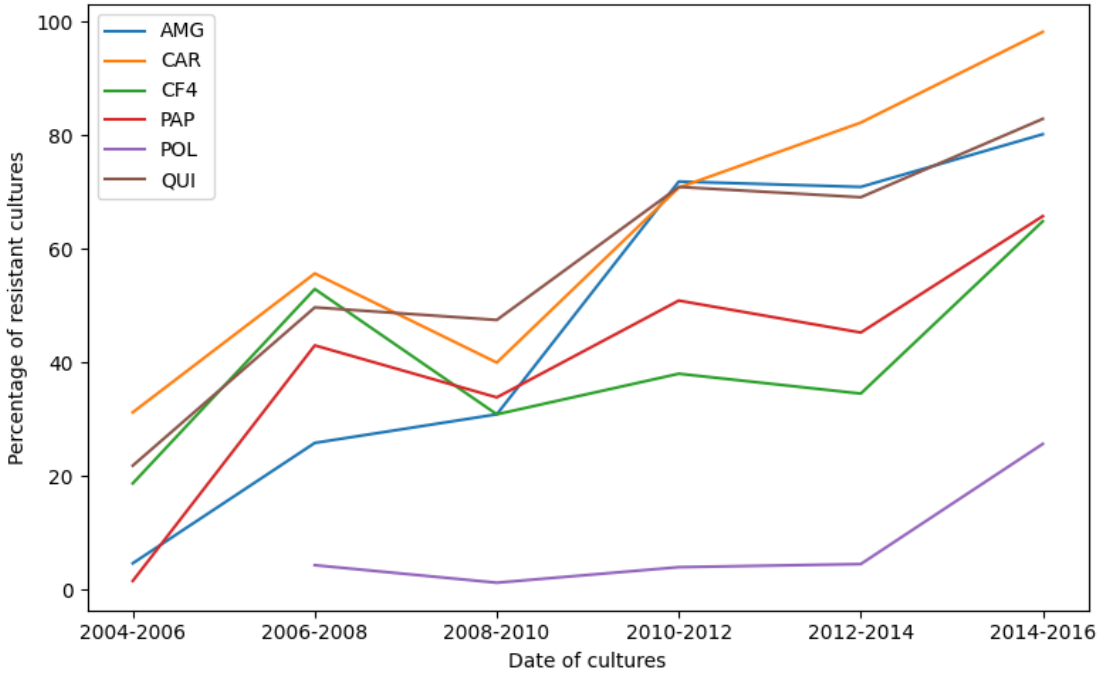
- For each experiment, a set of classifiers is trained, each for a different training-test window.
- The size chosen for test windows is fixed in 1 month, which is a relatively short time, very close to the training instances.
- Different test windows do not overlap between them. That is, compared to the others, each test window contains different instances belonging to different time intervals.
- Instances in the training window do not contain antibiograms belonging to patients that also are present in the test window. For instance, if the result of an antibiogram of a particular patient is to be predicted in the test set, there are not past antibiograms of the same patient in the training set. That way, it is ensured that patients from training and test windows are independent.

In all the experiments of this study, the training window will start containing the instances that belong to the time interval from 2004 to 2011 (8 years), and the test window will start by counting with the instances of the first month (January) of 2012. After that, the training window will increase in size to consider one more month (January of 2012) and, as previously described, the test window will shift a month to now consider just February of 2012. This goes on until all the data set is traversed. Thus, test windows traverse years from 2012 to 2016 (5 years). With

each training and test pair of windows, simple validation is performed to maintain the temporal order.

In the data set it is observed that as time goes by, more and more bacteria become antimicrobial resistant (which actually is the motivation of this work). This trend can be observed in Fig. 3. Each value in the plot represents the percentage of resistant cultures for 2 years and every antimicrobial family. Each antimicrobial family is depicted with a different color and it can be seen that the more time passes the more resistant bacteria become to any antibiotic. This causes that, at the beginning, the majority of instances are sensitive while at the end the majority are the resistant ones. With that, the *incremental training window* at first shows *class imbalance*, with the majority class being sensitive and, as it grows, it is less unbalanced because of the increase in the amount of resistant instances. Nevertheless, for most of the data sets, training windows are not fully balanced even when training window is almost as big as all the data set. On the other hand, since the test window has a fixed size, it is almost always *class unbalanced*.

Figure 3: Evolution of antimicrobial resistance over time



To get a realistic approximation of performance on the experiments, the true neg-

atives (success in sensitive instances) and the true positives (success in resistant instances) are calculated, together with the general accuracy. For a particular test window with n_s sensitive instances and n_r resistant instances, if the method succeeds in predicting p_s sensitive instances and p_r resistant instances, the just mentioned values are calculated as:

$$\text{general\_accuracy} = \frac{p_s + p_r}{n_s + n_r} \tag{3.4}$$

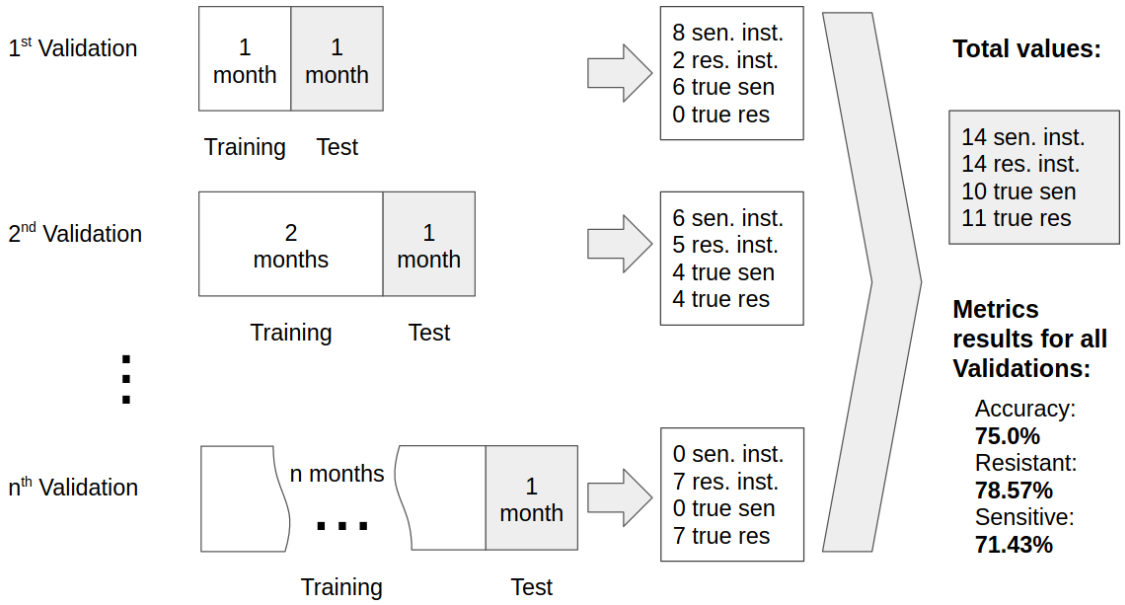
$$\text{resistant\_success} = \frac{p_r}{s_r} \tag{3.5}$$

$$\text{sensitive\_success} = \frac{p_s}{n_s} \tag{3.6}$$

These metrics offer a better approximation of the performance, because allow to track the success rate in the minority class label, which in many real problems is the most important one. For instance, if the test set counts with 8 sensitive and 2 resistant instances and the ML technique predicts all instances as sensitive, the general accuracy metric would have a pretty high value of a 80%, while it will be performing poorly in identifying resistant antibiograms which are the ones most needed to detect.

To calculate the mean accuracy among several windows, an accumulation of the success rates is done, and the performance is evaluated using Equations 3.4, 3.5 and 3.6. In other words, the performance of several windows is not calculated by averaging the individual performance values, but by accumulating the number of success instances in each window and using it to calculate the accuracy. This is done because test windows may have a different number of instances, due to the fact that not all 1-month time intervals have the same number of antibiograms. Therefore, making an average between their accuracy values would not be adequate since some instances would have more weight than others depending on the number of instances in their test window. Figure 4 presents a diagram of the *incremental training window* scheme with an example of its application.

Figure 4: Incremental training window scheme



On the left of the diagram it is depicted how the training window size increases step by step (from top to bottom), and how the test window shifts in each validation stage. In the middle there is an example of instances distribution and the number of right predictions, for each of the validation steps. On the right, mean values for the success rates are calculated by supposing that the nth validation is the third and last one. As described, different values are added for the total number of sensitive instances, the total number of resistant instances, the number of sensitive instances correctly classified and the number of resistant instances correctly classified. Finally success metrics are calculated by using Equations 3.4, 3.5 and 3.6.

In the second type of experiments, the effectiveness of the incremental window is tested. The data set used just considers the features initially given detailed in Table 1, and the generated features are not considered. The ML technique used for target's prediction is *Random Forest* (RF) with 100 estimators, and the nodes of the trees are expanded until all leaves are pure or until all leaves contain less than 2 samples. RF is selected because it is the method providing best results, as it is observed in Section 4.3. The first part of the experiment, measures the accuracy achieved by a training window with a fixed size of a year, and a test window of a fixed 1-month size is used by shifting windows one month at a time. Here, the distance between the train and test window is varied from 4 to 0 years. Finally, the outcomes produced by the *incremental training window* are compared to those of the training window with fixed size.

In order to assess the contribution of the proposed features to the prediction of the target, four kinds of experiments have been carried out. In the first one accuracy has been calculated using just the initial features of the data set without using the proposed features, in the second one $p\&*$ features have been considered together with the initial data set, in the third experiment $r\&*$ features have been considered together with the initial data set, and in the fourth one both $p\&*$ and $r\&*$ features have been used together with the initial data set. Every experiment considers the prediction for each of the six antimicrobial families, by using their respective data set and target feature. Also in each experiment, 5 different ML methods have been employed which are *Logistic Regression* (LR), *K-Nearest Neighbors* (KNN), RF, *Support Vector Machine* (SVM) and *Dynamic Selection* (DS).

In the case of DS, KNORA technique is used. In this work, a pool of 10 decision tree classifiers are trained to be used as base classifiers, and the training window is randomly dived in two halves, one to be used as training instances and the other as the validation set.

In this group of four experiments, hyperparameters of the ML methods are adjusted, and the ones reporting the best performance are selected. In the case of LR, the hyperparameter considered is the regularization parameter C, which is the inverse of regularization strength, smaller values specify stronger regularization. In KNN, the number of neighbors K is taken into account. In RF, are considered the number of estimators or trees in the forest n and the maximum depth of trees d. In SVM, as in LR, the regularization paramenter C is considered together with γ which is the kernel coefficient of the Radial Basis Function (RBF) kernel. Finally, in DS the size of the neighborhood k is taken into account as hyperparameter.

In the next group of experiments, feature selection is carried out since it has been shown that, in some cases, it improves accuracy, efficiency, applicability and understandability of a learning process and its resulting model [31]. The values provided by feature weighting in the first experiment are used as the feature evaluation measure, and the cutting criterion is based on selecting the number of features offering the best performance on prediction. The ML methods used are the ones giving

best results on the previous four experiments for each of the targets, with their hyperparameters properly tuned.

As it has been previously noticed, training windows suffer from *class imbalance*. There are some possibilities to overcome this problem. One approach would be to perform undersampling, reducing the number of instances in the majority class to the number of the minority class. Although in some cases it is useful, in this problem data is very limited, and reducing instances directly affects the performance of ML methods. For this reason, oversampling is applied to balance classes. Experiments carried out consist of applying oversampling through Borderline-SMOTE technique to balance classes (sensitive and resistant) on the *incremental training window* instances. Once classes are balanced, feature selection is performed, as before, using the ML methods providing best results on previous experiments.

Finally, a MDR classifier for *Pseudomonas aeruginosa* is built as an ensemble of the six classifiers trained to predict resistance for each of the antimicrobial families.

Chapter 4

Results

This chapter presents the results obtained after carrying out the experiments described in Section 3.4. In Section 4.1 the relevance of the set of features is calculated regarding each antimicrobial family. The *incremental training window scheme* is evaluated in Section 4.2. Section 4.3 assesses the utility of the 12 generated features. Using results of feature weighting, Sections 4.4 and 4.5 apply feature selection, together with oversampling in the case of Section 4.5. The evolution of success in prediction is expressed in Section 4.6. Finally, Section 4.7 shows the results of MDR prediction using an ensemble of classifiers.

4.1 Feature weighting

The results of feature relevance using the average of the three aforementioned feature weighting methods (IG, CVD and UEB-1) are displayed in six different tables, one per each target feature representing an antimicrobial family. Tables 4, 5, 6, 7, 8 and 9 show the resulting feature's weights for the target features `c&amg`, `c&car`, `c&cf4`, `c&pap`, `c&pol` and `c&qui` respectively.

It is noticeable that the most important feature for `c&amg`, `c&car`, `c&cf4`, `c&pap` and `c&qui` are `p&amg`, `p&car`, `p&cf4`, `p&pap` and `p&qui` respectively. That is, for each of those five targets, the most important feature is the one indicating the results

of past antibiograms performed to the patient, for the same antimicrobial family of the target. This makes sense from a clinical point of view. It denotes that, for instance, knowing whether a patient has previously been infected with a bacterium resistant to AMG helps predict if the patient is currently infected with a bacterium resistant to AMG. This occurs because, a patient that, during the last 21 days, has been infected with a resistant bacterium is very likely still infected.

In the case of `c&pol`, the most important feature is not `p&pol`, although it is fourth most important, and is the most important regarding `p&*` features.

Observing the relative position of the proposed features (`p&*` and `r&*`), one can note that they are among the most important features of the data set. In `c&amg`, `c&car`, `c&cf4`, `c&pap`, `c&pol` and `c&qui` the 12 proposed features are included within the 19, 21, 18, 19, 26 and 19 most important features, respectively.

As before, in the case of `c&pol` target, proposed features have the lowest importance compared to their relevance when considering the other targets. This is due to the fact that the vast majority of antibiogram results for POL are sensitive. This causes the majority of values for `p&pol` and `r&pol` to be 0, which does not provide much information, causing a reduction on their relevance. Furthermore, the rest of proposed features vary greatly while the class of `c&pol` is almost constant with sensitive value.

It can also be noticed that, in general, `p&*` features get higher weights than `r&*` features. This is probably caused by a more direct relation between previous infections of a patient and current infections of the same patient, than between previous infections in the ICU as a whole and current infections of a particular patient.

In addition to proposed features, `goi_A`, `goi_B`, `goi_D` and `goi_E` are some of the most relevant features. They are between the 10 most relevant features for all six targets. These features detail the comorbidities of patients dividing diseases in groups of illness. From a clinical perspective the importance of these four features is justified, since comorbidity is a known risk factor for antibiotic-resistant bacterial infections [32, 33, 34, 35].

Table 4: Feature weighting for c&amg target

Feature name	Weights for c&amg
p&amg	9.9889
goi_A	8.6435
number_antibiotics	8.5477
goi_B	8.52
p&pol	8.4569
p&qui	8.4034
goi_D	8.169
r&pol	8.0721
goi_E	8.0508
days_to_culture	7.6961
r&cf4	7.4636
r&amg	7.4611
culture_type_grouped	7.4525
p&cf4	7.3411
r&pap	7.3253
r&car	7.2563
p&car	7.2339
r&qui	7.1932
p&pap	7.1894
date_culture	7.1195
pluripathology	6.6048
age	6.0431
goi_F	5.7506
reason_admission	5.6675
goi_G	5.5764
culture_type	5.4247
goi_C	5.4177
patient_category	5.3412
origin	5.281
gender	5.2539
day_month_admission	5.0838
month_admission	4.9426
day_week_culture	4.9011
month_culture	4.8763
day_week_admission	4.6345
day_month_culture	4.0581
culture_type_grouped_2	3.5146

Table showing the weights given by the average of IG, CVD and UEB-1 to each of the features in descending order. Left column contains names of features and right column the values for AMG.

Table 5: Feature weighting for c&car target

Feature name	Weights for c&car
p&car	9.2378
goi_A	8.7422
goi_B	8.637
p&pol	8.6006
number_antibiotics	8.4841
p&qui	8.4739
goi_D	8.3529
goi_E	8.3246
goi_F	8.1353
r&pol	8.0593
days_to_culture	7.8243
p&amg	7.574
r&cf4	7.4225
culture_type_grouped	7.4134
r&amg	7.4108
r&pap	7.3598
r&car	7.3101
date_culture	7.0660
r&qui	7.0107
p&cf4	6.9813
p&pap	6.8552
pluripathology	6.5512
age	6.1329
origin	5.9304
goi_G	5.6799
day_week_culture	5.5983
gender	5.5712
reason_admission	5.4169
goi_C	5.3855
culture_type	5.285
day_month_admission	5.2647
month_admission	5.0551
month_culture	5.0499
day_week_admission	5.0333
patient_category	4.8038
day_month_culture	4.1842
culture_type_grouped_2	3.6857

Table showing the weights given by the average of IG, CVD and UEB-1 to each of the features in descending order. Left column contains names of features and right column the values for CAR antimicrobial family.

Table 6: Feature weighting for c&cf4 target

Feature name	Weights for c&cf4
p&cf4	10.0
p&pap	8.9153
p&qui	8.5855
p&pol	8.5642
goi_A	8.53
goi_B	8.4005
goi_E	8.2275
goi_D	8.1139
r&pol	8.0542
days_to_culture	7.8426
p&amg	7.5361
culture_type_grouped	7.4415
r&cf4	7.4275
r&pap	7.4169
r&car	7.2667
r&amg	7.2101
p&car	7.1701
r&qui	6.9297
pluripathology	6.6813
number_antibiotics	6.2935
age	6.1125
date_culture	6.0097
goi_F	5.9607
reason_admission	5.6289
goi_G	5.5155
culture_type	5.5107
goi_C	5.3763
gender	5.2892
day_week_admission	5.2768
day_month_admission	5.2638
origin	5.1825
month_admission	5.0757
day_week_culture	5.0218
month_culture	5.0027
patient_category	5.0001
day_month_culture	4.4134
culture_type_grouped_2	3.8370

Table showing the weights given by the average of IG, CVD and UEB-1 to each of the features in descending order. Left column contains names of features and right column the values for CF4 antimicrobial family.

Table 7: Feature weighting for c&pap target

Feature name	Weights for c&pap
p&pap	9.8534
p&cf4	9.7292
p&pol	9.1345
goi_A	9.0537
goi_B	8.9282
p&qui	8.9018
goi_D	8.6165
r&pol	8.591
goi_E	8.5319
days_to_culture	8.2969
p&amg	8.0266
r&pap	7.9072
r&cf4	7.8779
culture_type_grouped	7.8636
r&car	7.6772
r&amg	7.675
p&car	7.4589
number_antibiotics	7.3668
r&qui	7.3267
pluripathology	6.9773
age	6.3819
goi_F	6.2847
date_culture	6.2203
goi_G	5.8612
reason_admission	5.841
culture_type	5.7344
gender	5.6496
patient_category	5.6372
goi_C	5.6159
day_week_culture	5.5133
day_month_admission	5.4706
day_week_admission	5.4293
origin	5.4243
month_admission	5.2573
month_culture	5.2035
day_month_culture	4.7819
culture_type_grouped_2	3.8929

Table showing the weights given by the average of IG, CVD and UEB-1 to each of the features in descending order. Left column contains names of features and right column the values for PAP antimicrobial family.

Table 8: Feature weighting for c&pol target

Feature name	Weights for c&pol
goi_A	9.3524
goi_B	9.0552
goi_D	8.9492
p&pol	8.9277
goi_E	8.7444
goi_G	8.4566
r&pol	8.4004
days_to_culture	8.323
culture_type_grouped	7.6842
r&cf4	7.4217
r&pap	7.3161
pluripathology	7.174
number_antibiotics	7.0749
date_culture	7.0528
r&amg	6.9603
gender	6.9412
r&car	6.8556
p&cf4	6.7566
p&amg	6.6943
r&qui	6.4349
month_admission	6.3505
goi_F	6.3328
origin	6.2911
p&car	6.2295
p&qui	6.2281
p&pap	6.2255
day_week_culture	6.1982
reason_admission	6.0867
age	6.0319
goi_C	6.0069
month_culture	5.8353
patient_category	5.5646
day_week_admission	5.4291
day_month_admission	5.4279
culture_type	5.2893
day_month_culture	4.3751
culture_type_grouped_2	3.9667

Table showing the weights given by the average of IG, CVD and UEB-1 to each of the features in descending order. Left column contains names of features and right column the values for POL antimicrobial family.

Table 9: Feature weighting for c&qui target

Feature name	Weights for c&qui
p&qui	10.0
p&amg	9.0248
p&car	8.2633
p&pol	7.9879
goi_A	7.966
goi_B	7.8204
goi_D	7.5087
r&pol	7.4819
p&cf4	7.4741
goi_E	7.4234
days_to_culture	7.3608
p&pap	7.0711
number_antibiotics	7.027
r&cf4	7.0199
r&amg	6.9423
r&pap	6.9278
culture_type_grouped	6.8836
r&car	6.8781
r&qui	6.7281
pluripathology	6.2834
age	6.2379
date_culture	5.9938
patient_category	5.462
goi_F	5.4429
reason_admission	5.3852
goi_G	5.2014
culture_type	5.1842
goi_C	5.1506
month_admission	5.1472
gender	5.0189
origin	4.9419
day_month_admission	4.9023
day_week_culture	4.7951
month_culture	4.7261
day_week_admission	4.5163
day_month_culture	4.0241
culture_type_grouped_2	3.352

Table showing the weights given by the average of IG, CVD and UEB-1 to each of the features in descending order. Left column contains names of features and right column the values for QUI antimicrobial family.

4.2 Incremental training window evaluation

Before assessing ML methods and features capability to predict *Pseudomonas aeruginosa* sensitivity, it is evaluated whether the *incremental training window* is a good scheme for prediction. Results are presented in Table 10.

Table 10: Incremental window evaluation

Family	Metric	4	3	2	1	0	Incremental
AMG	accuracy	47.093	47.1264	50.289	61.5385	66.4921	68.1818
	resistant	37.2263	45.7143	49.6241	68.8312	76.5101	76.129
	sensitive	85.7143	52.9412	52.5	34.1463	30.9524	39.5349
CAR	accuracy	54.3046	62.9139	67.5497	80.4511	89.5522	85.0299
	resistant	51.7483	63.6364	70.0	85.2459	94.4882	89.1026
	sensitive	100.0	50.0	36.3637	27.2727	0.0	27.2727
CF4	accuracy	45.8763	47.4286	54.0323	54.5455	52.9412	54.5455
	resistant	2.0202	3.2258	3.5714	46.4789	46.5909	42.1569
	sensitive	91.5789	97.561	95.5882	62.5	59.7561	67.7083
PAP	accuracy	45.6853	43.5897	44.3182	46.3855	54.6448	45.9184
	resistant	16.6667	9.8214	6.3158	50.5495	62.5	44.2478
	sensitive	85.5422	89.1566	88.8889	41.3333	44.3038	48.1928
POL	accuracy	-	-	-	-	66.6667	80.1653
	resistant	-	-	-	-	50.0	17.8571
	sensitive	-	-	-	-	75.0	98.9247
QUI	accuracy	42.7136	51.5464	47.3684	66.1538	74.8691	73.7374
	resistant	34.8101	43.2258	48.9933	72.2581	89.404	83.5443
	sensitive	73.1707	84.6154	41.4634	42.5	20.0	35.0

Table representing the accuracy achieved with different set-ups of training windows, for each of the antimicrobial families. Values in columns from 4 to 0 express the number of years of distance between a 1-year fixed size training window and the test window. Incremental column show the results obtained by using the incremental training window. The ML method applied in these experiments is RF, and the data set just considers initial features.

First, training windows with one-year fixed size are used in a set of experiments. Different experiments vary the distance between training and test window to observe the change in performance this causes. It can be noticed that when training window is 4 years apart from the test window, the general accuracy of RF is worse than random (under 50%). This happens for every antimicrobial family except for CAR, whose result is slightly better than random, and POL that don't have values for 4,

3, 2 and 1 years of distance because training windows generated in this manner have too few resistant instances to train the classifier.

The general trend observed is that accuracy metric improves as the training window gets closer to the test window, resulting in higher accuracy values when distance is 0 years, that is, when training is next to test. These results supports the hypothesis exposed in Section 3.4, changes in the hidden context are produced in the data set overtime and they can induce changes in the target concept.

Success on resistant instances increases as training window approaches the test window, in the same way as the accuracy metric. On the other hand, sensitive success starts with a high percentage for all families, and it decreases as getting close to the 0 value for years of distance. The high percentage at the beginning is due to the fact that, when training window is far from the test, it gets closer to the beginning of the data set, where the majority class is sensitive, thus producing bias towards predicting instances as sensitive. If the majority of test instances are classified as sensitive, most of the true sensitive ones will be correctly predicted but it will fail on classifying resistant instances as it is observed in the column 4. Resistant success decreases when approaching 0 distance because the classifier is no longer biased towards sensitive instances, making it trickier to predict them with less training sensitive instances and, furthermore, now it is a little biased towards resistant instances. Either way, results obtained with 0 years are, in general, the best ones because their accuracy is the highest, making them better than random.

Looking now at the performance of the proposed *Incremental window* scheme, it is observed that the sensitive success percentage is in all cases higher than the sensitive success percentage in the "0" column, this is done by considering instances from the beginning of the data set, solving in that way, the problem of not having enough sensitive instances. In addition, accuracy metric and resistant success using the *Incremental window* are the among the highest of the data set. Although performance for PAP and POL antimicrobial families is not as good as expected, in the case of the first one, values from resistant success and sensitive success are balanced and, in the case of POL, the accuracy is relatively high.

Incremental window is, in general, the scheme providing the best balance between the three different metrics and, therefore, it is the approach used in the rest of the experiments of the work.

4.3 Generated features evaluation

To test whether proposed features provide useful information for prediction, a set of ML methods is trained for each of the antimicrobial families, using 4 different configurations for the data sets. The four configurations consist on: considering just the features of the given initial data set, considering proposed `p&*` features together with the initial features, considering proposed `r&*` features together with the initial features, and considering both `p&*` and `r&*`. Also, the hyperparameters of each ML method are tuned so that the ones providing best results are kept. Taking into account the four possible configurations generated by using `p&*` and `r&*` features and testing with them all selected ML methods, it is ensured to find the configuration, among the possible four, performing the best for every antimicrobial family.

Results of experiments that take into account the first configuration, in which just features of the initial data set are used, are shown in Table 11, and the respective hyperparameters for each method and antimicrobial family, are detailed in Table 12. The percentages considered best for a particular antimicrobial family are those with a better balance among the three metrics. For instance, in the case of AMG family, LR achieves an accuracy of a 79.798%, but the SVM result with an accuracy of 74.2424% is considered better because resistant and sensitive success values have a more similar percentage, in this case this means that the sensitive success rate has a higher value in SVM than in LR. It can be observed that SVM method provides the best outcomes for all families except for QUI, for which best outcomes are achieved with RF. It is remarkable that the best performance values for this first configuration are already better than the ones in the previous experiments where the *Incremental training window* scheme is tested with RF.

In the case of the second configuration, when adding `p$*` features to the initial ones,

Table 11: Results using the initial data set features

Family	Metric	LR	KNN	RF	SVM	DS
AMG	accuracy	79.798	72.2222	68.1818	**74.2424**	61.1111
	resistant	100.0	83.2258	76.129	**79.3548**	67.0968
	sensitive	6.9767	32.5581	39.5349	**55.814**	39.5349
CAR	accuracy	94.012	89.8204	86.8263	**84.4311**	86.8263
	resistant	100.0	94.2308	91.0256	**86.5385**	91.6667
	sensitive	9.0909	27.2727	27.2727	**54.5455**	18.1818
CF4	accuracy	56.5657	65.1515	55.0505	**66.6667**	55.0505
	resistant	36.2745	74.5098	42.1569	**76.4706**	49.0196
	sensitive	78.125	55.2083	68.75	**56.25**	61.4583
PAP	accuracy	52.0408	59.6939	47.9592	**61.2245**	48.4694
	resistant	78.7611	73.4513	44.2478	**69.0265**	45.1327
	sensitive	15.6627	40.9639	53.012	**50.6024**	53.012
POL	accuracy	76.8595	84.2975	80.1653	**85.124**	77.686
	resistant	0.0	57.1429	17.8571	**60.7143**	7.1429
	sensitive	100.0	92.4731	98.9247	**92.4731**	98.9247
QUI	accuracy	79.798	76.7677	**76.2626**	72.7273	67.1717
	resistant	99.3671	92.4051	**86.0759**	81.6455	74.6835
	sensitive	2.5	15.0	**37.5**	37.5	37.5

Table showing the success metrics for each ML method and antimicrobial family. It uses just the initial features selected in the data set, not considering the generated features. Best results for each family are marked in bold.

Table 12: Hyperparameters for initial data-set features

Model	Hyperparameter	AMG	CAR	CF4	PAP	POL	QUI
LR	C	1	1	10	10	1	0.001
KNN	K	3	3	14	14	3	2
RF	n	100	200	100	200	100	200
	d	20	20	20	10	20	10
SVM	C	10000	10000	10000	1	100	10000
	γ	1e-05	1e-06	1e-07	1e-06	1e-05	1e-06
DS	k	23	2	8	7	3	2

it is observed a significant improvement of performance, for almost all antimicrobial families. Values obtained for this configuration and their hyperparameters are in Table 13 and Table 14, respectively. Now, the best performing methods are RF for AMG, CF4, PAP and QUI, SVM is best for POL and DS for CAR. The families having their results improved with respect to the first configuration are AMG, CAR,

CF4 and QUI, and POL remains with the same value. Furthermore, all of the metrics for the best values of each family have a percentage higher than 50%. These results denote that `p$*` features actually improve classification of antibiograms, as it was previously suggested by feature weighting.

Table 13: Results using `p&*` features together with initial data set features

Family	Metric	LR	KNN	RF	SVM	DS
AMG	accuracy	77.2727	72.2222	**78.2828**	73.2323	69.1919
	resistant	96.7742	83.2258	**83.871**	78.7097	75.4839
	sensitive	6.9767	32.5581	**58.1395**	53.4884	46.5116
CAR	accuracy	95.2096	89.8204	88.024	83.8323	**89.2216**
	resistant	100.0	94.2308	91.0256	85.8974	**91.0256**
	sensitive	27.2727	27.2727	45.4545	54.5455	**63.6364**
CF4	accuracy	59.0909	65.1515	**70.2020**	66.6667	63.1313
	resistant	42.1569	74.5098	**60.7843**	76.4706	61.7647
	sensitive	77.0833	55.2083	**80.2083**	56.25	64.5833
PAP	accuracy	53.0612	59.6939	**60.2041**	61.2245	55.102
	resistant	79.646	73.4513	**56.6372**	69.0265	55.7522
	sensitive	16.8675	40.9639	**65.0602**	50.6024	54.2169
POL	accuracy	76.8595	84.2975	77.686	**85.124**	78.5124
	resistant	0.0	57.1429	3.5714	**60.7143**	10.7143
	sensitive	100.0	92.4731	100.0	**92.4731**	98.9247
QUI	accuracy	80.303	76.7677	**80.8081**	72.7273	76.2626
	resistant	97.4684	92.4051	**87.3418**	80.3797	81.0127
	sensitive	12.5	15.0	**55.0**	42.5	57.5

Table showing the success metrics for each ML method and antimicrobial family. It uses `p&*` features together with initial features in the data set. Best results for each family are marked in bold.

Table 14: Hyperparameters for `p&*` features and initial data-set features

Model	Hyperparameter	AMG	CAR	CF4	PAP	POL	QUI
LR	C	0.1	0.1	10	0.1	1	100
KNN	K	3	3	14	14	3	2
RF	n	100	300	100	300	100	100
	d	20	10	30	20	20	30
SVM	C	10000	10000	10000	1	100	100000
	γ	1e-05	1e-06	1e-07	1e-06	1e-05	1e-07
DS	k	22	2	2	9	2	5

The performance of the third configuration, in which `r&*` features are added to

initial features, is represented in Table 15 with its respective hyperparameters in Table 16. It shows similar results as the ones seen for the first configuration. Again the best ML method for all families is SVM except for QUI which gets better results with RF. Although in general this configuration is not better than the first one, it is noticed just a slight improvement on prediction of QUI, with the sensitive success rate increasing from a 37.5% in the first to a 45.0% in the current configuration, making sensitive and resistant success rates more balanced while maintaining the same accuracy.

Table 15: Results using `r&*` features together with initial data set features

Family	Metric	LR	KNN	RF	SVM	DS
AMG	accuracy	79.798	72.2222	73.7374	**72.7273**	68.1818
	resistant	100.0	83.2258	83.2258	**79.3548**	77.4194
	sensitive	6.9767	32.5581	39.5349	**48.8372**	34.8837
CAR	accuracy	94.012	89.8204	89.2216	**82.6347**	84.4311
	resistant	100.0	94.2308	94.2308	**83.9744**	87.8205
	sensitive	9.0909	27.2727	18.1818	**63.6364**	36.3636
CF4	accuracy	55.0505	65.1515	52.0202	**66.6667**	50.5051
	resistant	32.3529	74.5098	39.2157	**77.451**	45.098
	sensitive	79.1667	55.2083	65.625	**55.2083**	56.25
PAP	accuracy	53.0612	59.6939	48.9796	**61.2245**	48.4694
	resistant	82.3009	73.4513	46.0177	**69.0265**	43.3628
	sensitive	13.2530	40.9639	53.012	**50.6024**	55.4217
POL	accuracy	76.8595	84.2975	80.1653	**85.124**	77.686
	resistant	0.0	57.1429	25.0	**60.7143**	17.8571
	sensitive	100.0	92.4731	96.7742	**92.4731**	95.6989
QUI	accuracy	79.798	76.7677	**76.2626**	72.7273	65.6566
	resistant	99.3671	92.4051	**84.1772**	81.0127	74.0506
	sensitive	2.5	15.0	**45.0**	40.0	32.5

Table showing the success metrics for each ML method and antimicrobial family. It uses `r&*` features together with initial features in the data set. Best results for each family are marked in bold.

The last configuration results and hyperparameters are explicited in Table 17 and Table 18, respectively. In this set-up both `p&*` and `r&*` features are added to initial features. The best performing methods are DS for AMG, RF for CAR, CF4, PAP and QUI and SVM for POL. It is observed that now, performance of CAR, PAP and QUI families is improved with respect to all three previous configurations. POL

Table 16: Hyperparameters for `r&*` features and initial data-set features

Model	Hyperparameter	AMG	CAR	CF4	PAP	POL	QUI
LR	C	10	1	1	0.001	1	0.001
KNN	K	3	3	14	14	3	2
RF	n	300	100	200	50	50	100
	d	20	10	20	30	10	20
SVM	C	10000	100000	10	1	100	10000
	γ	1e-05	1e-06	1e-06	1e-06	1e-05	1e-06
DS	k	7	15	3	2	2	3

gets again the same best results as before. With this, it is inferred that `r&*` features actually contain some helpful information for prediction of the target.

It is worth mentioning that KNN produces the same results for any of the four configurations. Also, although LR gets high accuracy values in all configurations, success rates for sensitive and resistant instances are very unbalanced.

Table 19 is a summary of the best results for the different previous configurations an each antimicrobial family. It can be noticed that the best overall values for AMG and CF4 are achieved using `p&*` features, and the best results for CAR, PAP and QUI are obtained using both `p&*` and `r&*`. The best values for POL are equal independently from the configuration used. With that, it is noticed that both proposed features contain valuable information to improve performance, and although `p&*` features are specially effective, the set of features `r&*` also contribute on enhancing prediction.

4.4 Feature Selection

After assessing the benefits provided by proposed features, feature selection is used to discard unnecessary features.To perform feature selection it is used the ordering of features previously obtained by feature weighting. After that, success metrics are calculated for each antimicrobial family, varying the number of features from 1 to 37, which is the maximum number of features not counting the target feature. The minimum number of features returning best success metrics is selected. The ML methods used are the ones showing best performance, for each of the families, in the

Table 17: Results using `p&*` and `r&*` features together with initial data set features

Family	Metric	LR	KNN	RF	SVM	DS
AMG	accuracy	78.2828	72.2222	75.2525	73.2323	**76.7677**
	resistant	98.0645	83.2258	81.9355	80.0	**81.9355**
	sensitive	6.9767	32.5581	51.1628	48.8372	**58.1395**
CAR	accuracy	94.012	89.8204	**92.2156**	85.6287	89.2216
	resistant	98.0769	94.2308	**93.5897**	87.8205	91.0256
	sensitive	36.3636	27.2727	**72.7272**	54.5455	63.6364
CF4	accuracy	57.0707	65.1515	**67.1717**	66.6667	62.6263
	resistant	38.2353	74.5098	**58.8235**	77.451	54.902
	sensitive	77.0833	55.2083	**76.0417**	55.2083	70.8333
PAP	accuracy	56.1224	59.6939	**63.7755**	60.7143	59.1837
	resistant	79.646	73.4513	**67.2566**	68.1416	62.8319
	sensitive	24.0964	40.9639	**59.0361**	50.6024	54.2169
POL	accuracy	76.8595	84.2975	77.686	**85.124**	76.8595
	resistant	0.0	57.1429	3.5714	**60.7143**	10.7143
	sensitive	100.0	92.4731	100.0	**92.4731**	96.7742
QUI	accuracy	80.3030	76.7677	**79.798**	72.7273	74.7475
	resistant	98.1013	92.4051	**84.1772**	81.6456	80.3797
	sensitive	10.0	15.0	**62.5**	37.5	52.5

Table showing the success metrics for each ML method and antimicrobial family. It uses `p&*` and `r&*` features together with initial features in the data set. Best results for each family are marked in bold.

Table 18: Hyperparameters for `p&*` and `r&*` features and initial data-set features

Model	Hyperparameter	AMG	CAR	CF4	PAP	POL	QUI
LR	C	1	100	0.1	1000	1	100
KNN	K	3	3	14	14	3	2
RF	n	200	200	300	200	200	200
	d	20	20	30	20	10	20
SVM	C	10000	10000	10	1	100	10000
	γ	1e-05	1e-06	1e-06	1e-06	1e-05	1e-06
DS	k	17	3	8	23	2	2

last experiments.

Looking at Table 20 it is observed that all results are improved except for CAR, which needs all 37 features to get a high performance. AMG and POL maintain the same success percentages as before, but gets better by reducing the complexity of their models, now they require 30 and 29 features respectively. CF4, PAP and

Table 19: Summary of best results for each configuration of features

Family	Metric	Initial	`p&*`	`r&*`	`p&*` and `r&*`
AMG	accuracy	74.2424	**78.2828**	72.7273	76.7677
	resistant	79.3548	**83.871**	79.3548	81.9355
	sensitive	55.814	**58.1395**	48.8372	58.1395
CAR	accuracy	84.4311	89.2216	82.6347	**92.2156**
	resistant	86.5385	91.0256	83.9744	**93.5897**
	sensitive	54.5455	63.6364	63.6364	**72.7272**
CF4	accuracy	66.6667	**70.2020**	66.6667	67.1717
	resistant	76.4706	**60.7843**	77.451	58.8235
	sensitive	56.25	**80.2083**	55.2083	76.0417
PAP	accuracy	61.2245	60.2041	61.2245	**63.7755**
	resistant	69.0265	56.6372	69.0265	**67.2566**
	sensitive	50.6024	65.0602	50.6024	**59.0361**
POL	accuracy	**85.124**	**85.124**	**85.124**	**85.124**
	resistant	**60.7143**	**60.7143**	**60.7143**	**60.7143**
	sensitive	**92.4731**	**92.4731**	**92.4731**	**92.4731**
QUI	accuracy	76.2626	80.8081	76.2626	**79.798**
	resistant	86.0759	87.3418	84.1772	**84.1772**
	sensitive	37.5	55.0	45.0	**62.5**

The table shows the best results found in previous experiments for each configuration of features and antimicrobial family. Best results for each family are marked in bold.

QUI improve both their success metrics and dimensionality. In the case of CF4 it is notorious the small amount of features (3), it needs to get a good accuracy.

Thus, discarding features which probably introduce noise, not only reduction of complexity is acquired, but also an increment of success values.

Table 20: Results of feature selection

Family	Num. features	Success metrics		
		accuracy	resistant	sensitive
AMG	30	**78.2828**	**83.871**	**58.1395**
CAR	32	91.6168	92.9487	72.7273
CF4	3	**71.7172**	**62.7451**	**81.25**
PAP	20	**68.8776**	**68.1416**	**69.8795**
POL	29	**85.124**	**60.7143**	**92.4731**
QUI	32	**81.8182**	**84.8101**	**70.0**

Table showing the success metrics for each of the antimicrobial families after applying feature selection.

4.5 Oversampling and feature selection

In order to avoid bias towards a particular class, the classes of instances in training windows are balanced. To do that, the instances in the minority class are oversampled. Once this is done, feature selection is performed in the same way as in the last experiment and again, the best ML methods are used for prediction. Feature selection is also applied in this experiment because it has shown to improve results.

Table 21 demonstrate how oversampling together with feature selection reduces the dimensionality of the problem with respect to the previous experiment, for all the antimicrobial families, except for QUI that continues requiring 32 features. The families that further improve their success metrics with this scheme are CF4, PAP and QUI. It is remarkable how CF4 achieves its best performance just using the most relevant feature which is p&cf4 as shown in Table 6.

Although oversampling does not increase success for all families, it is still the best approach for half of them.

Table 21: Results of oversampling with feature selection

Family	Num. features	Value		
		accuracy	resistant	sensitive
AMG	29	78.2828	85.1613	53.4884
CAR	27	91.6168	93.5897	63.6364
CF4	1	**74.7475**	**62.7451**	**87.5**
PAP	15	**70.4082**	**65.4867**	**77.1084**
POL	27	76.8595	64.2857	80.6452
QUI	32	**82.8283**	**86.0759**	**70.0**

Table showing the success metrics for each of the antimicrobial families after applying oversampling followed by feature selection.

4.6 Temporal evolution of prediction

Using trained classifiers showing the best performance so far, it is observed how prediction accuracy evolves as a function of time. To briefly exemplify this, just AMG and QUI families have been selected. In Figure 5 it is shown how success

metrics evolve for AMG, starting at the point when training window just counts with 1 month of data. It can be observed that, at the beginning of the plot, sensitive success is very high while resistant success is very low. As it is noticed in Figure 6, this is due to the fact that the first instances of the data set, are mostly sensitive with a smaller amount of resistant instances. As time goes by, the number of sensitive instances is reduced as opposed to resistant instances, which makes the sensitive success decrease and resistant success increase. Nevertheless, the more time passes, the more the classifier learns by increasing the training window size, and sensitive and resistant metrics are closer to each other, which is a desirable effect. In the case of QUI, a similar behaviour is observed. As it is seen in Figure 7, accuracy grows as time passes by, which may indicate that the algorithm improves as it gets more training instances. Again, it is observed that resistant success overcomes sensitive success, as resistant become the predominant class Figure 8, although in this cases both success metrics are very close.

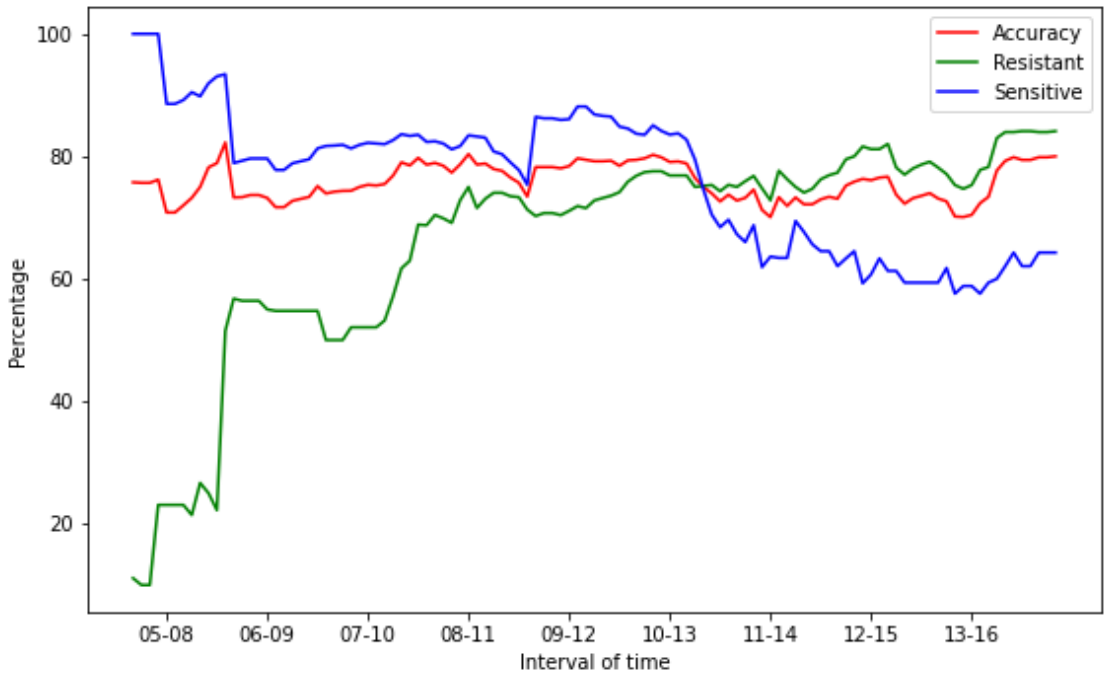
Each point in the plot is the mean of three years because, due to data scarcity and the uneven distribution of instances along time, small periods of time make success metrics drastically vary, resulting in a difficult visualization. In these figures it can be observed why the chosen test years ranges from 2012 to 2016; it is the interval where metrics are more stable and also predictors have a considerable amount of instances to learn. In the figures these years would be considered between $09 - 12$ and beyond $13 - 16$ in x axis values.

4.7 MDR predictor

Finally, Table 22 presents the best classifier's results achieved along the study. With these classifiers, an ensemble of predictors is built to predict MDR resistance for *Pseudomonas aeruginosa*.

The MDR predictor classifies as MDR antibiograms that are classified as resistant by three or more classifiers of the ensemble, and it classifies them as non-MDR otherwise.

Figure 5: Temporal evolution of `c&amg` prediction.



Three success metrics are displayed: accuracy, resistant success and sensitive success. The values in the x axis indicate the time interval comprised by each point in the plot. The first value in the axis is $(05-08)$, it denotes that its percentage value is the mean value of three years, from January 2005 to January 2008. The next point in the plot represents data from February 2005 to February 2008, and it goes on until the next x axis value which is $(06-09)$, meaning that it considers data from January 2006 to January 2009.

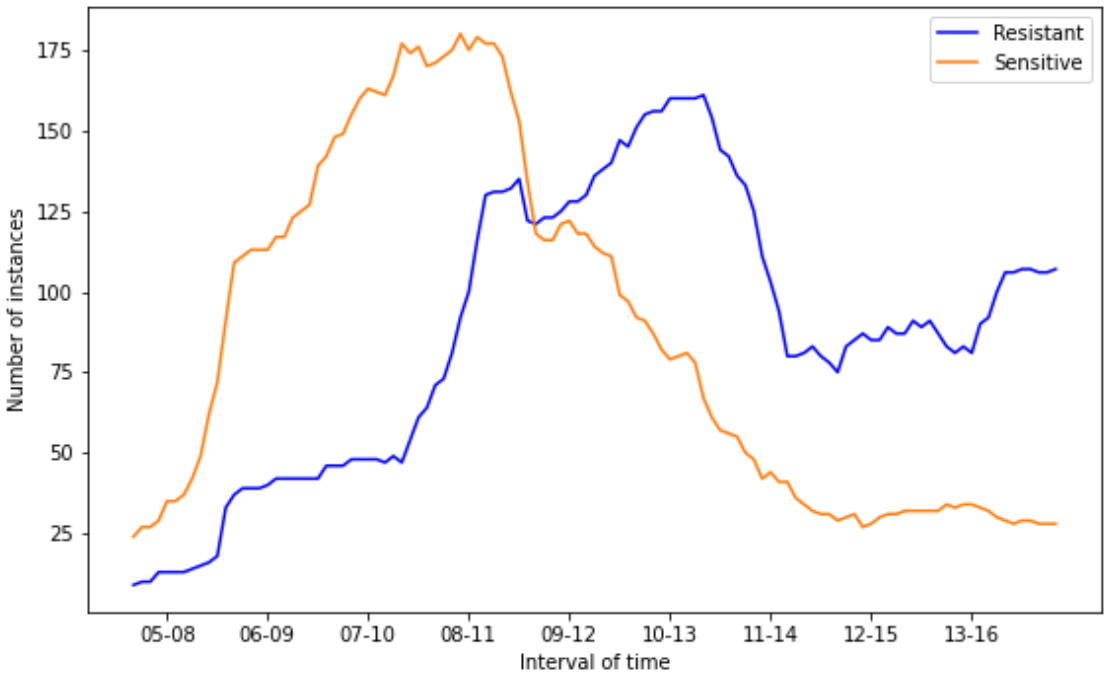
The final accuracy for the MDR predictor is 75.0%, with values for resistant and sensitive metrics of 78.67% and 56.67% respectively.

Table 22: Results of MDR predictor with optimal trained classifiers

Classifier	Accuracy	Resistant	Sensitive
AMG	78.2828	83.871	58.1395
CAR	92.2156	93.5897	72.7272
CF4	74.7475	62.7451	87.5
PAP	70.4082	65.4867	77.1084
POL	85.124	60.7143	92.4731
QUI	82.8283	86.0759	70.0
MDR	75.0	78.6667	56.6667

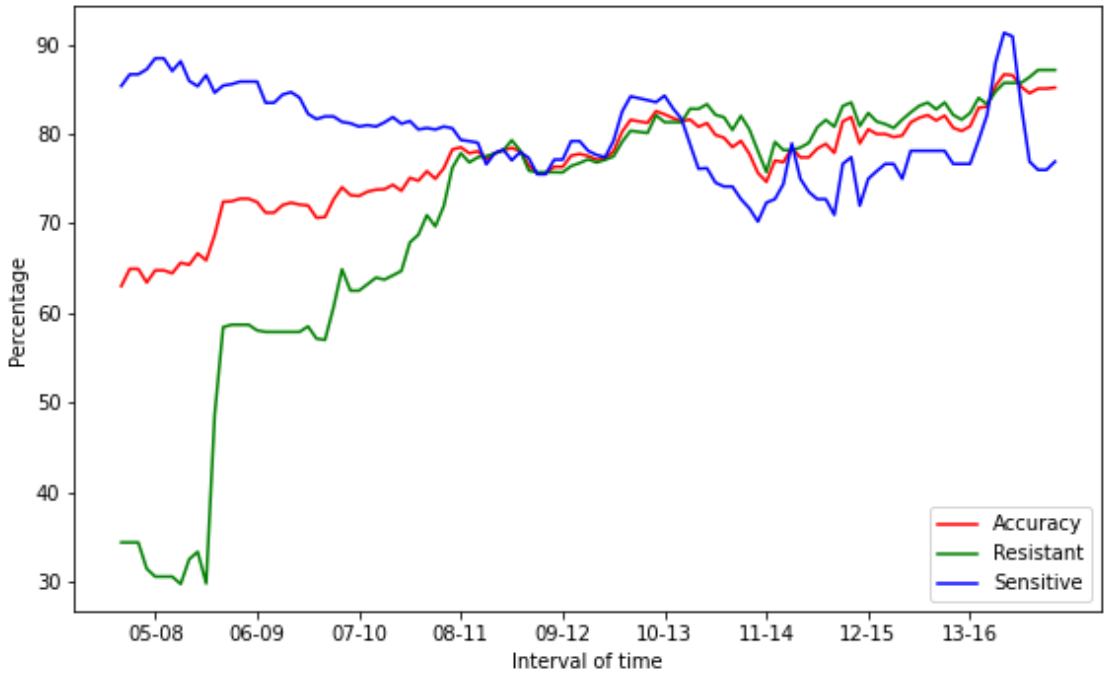
In the first part of the table, the success metrics for the best performing set-ups in the study are summarized. The MDR predictor generated as an ensemble of the best classifiers, provides the performance expressed in the last row of the table.

Figure 6: Temporal evolution of instances's classes for `c&amg` data set.



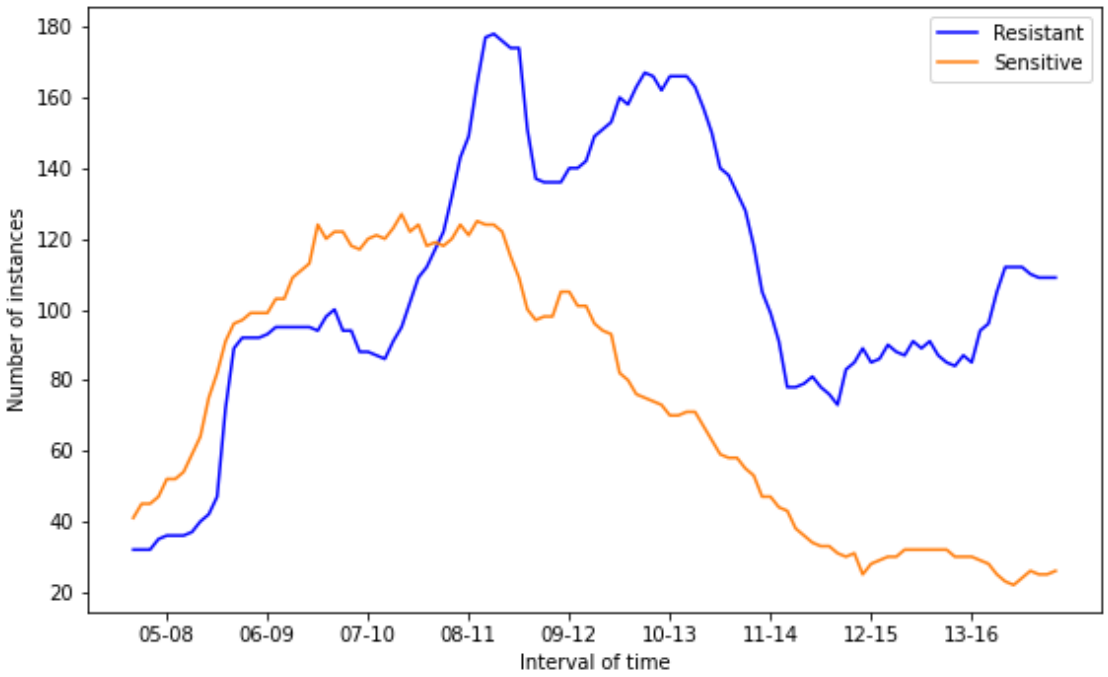
The amount of sensitive and resistance instances is expressed as a function of periods of time. The x axis values are explained in Figure 5 description.

Figure 7: Temporal evolution of `c&qui` prediction.



Three success metrics are displayed: accuracy, resistant success and sensitive success. The x axis values are explained in Figure 5 description.

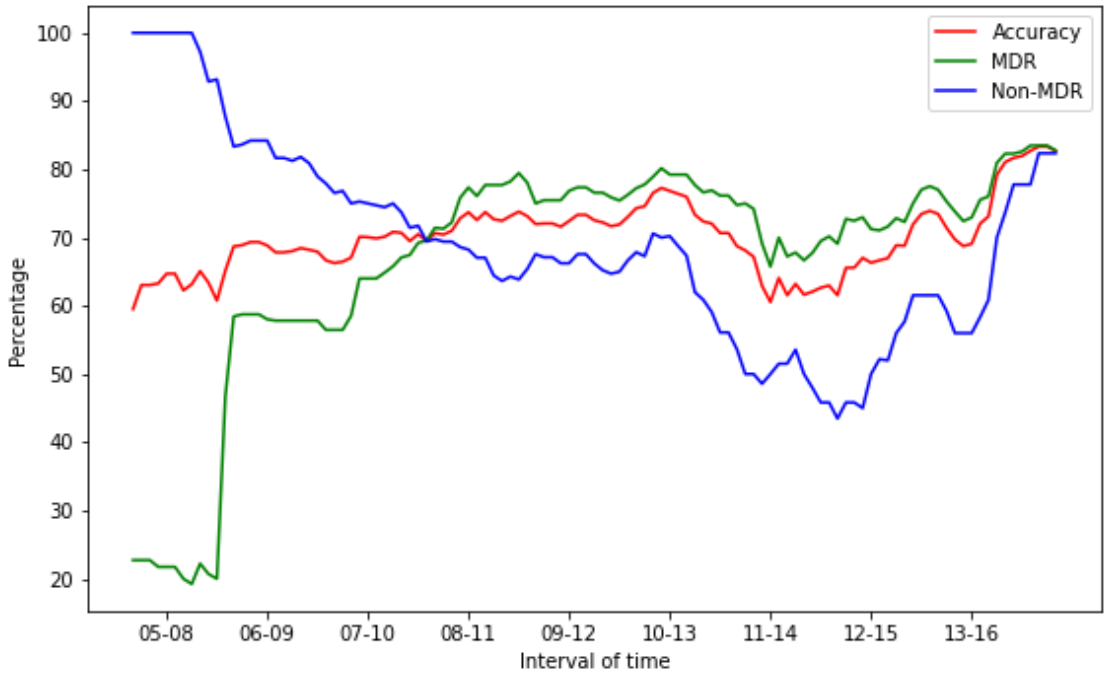
Figure 8: Temporal evolution of instances's classes for `c&qui` data set.



The amount of sensitive and resistance instances is expressed as a function of periods of time. The x axis values are explained in Figure 5 description.

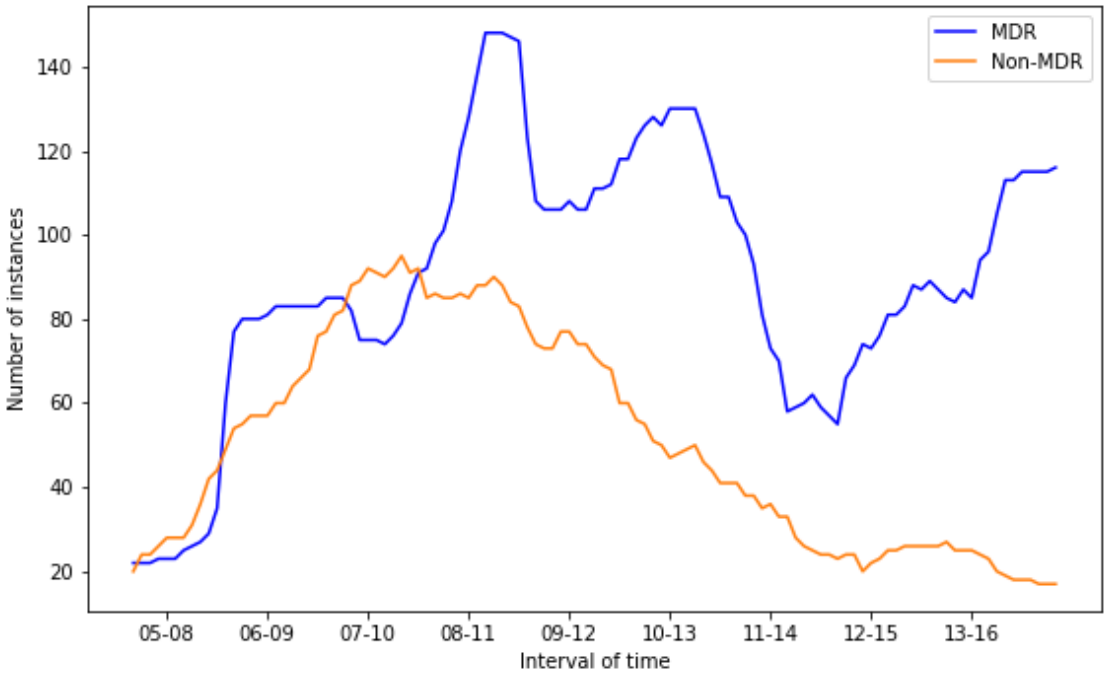
In Figures 9 and 10, the temporal evolution of MDR prediction is depicted. Its general trend is very similar to AMG and QUI temporal evolution of prediction performance. In this case, accuracy also appears to increase with more instances, ending in the last point with an accuracy of 82.71%, resistant success of 82.76% and sensitive success of 82.35%.

Figure 9: Temporal evolution of MDR prediction.



Three success metrics are displayed: accuracy, MDR success and Non-MDR success. The x axis values are explained in Figure 5 description.

Figure 10: Temporal evolution of instances's classes for MDR data set.



The amount of Non-MDR and MDR instances is expressed as a function of periods of time. The x axis values are explained in Figure 5 description.

Chapter 5

Discussion

In this work it is suggested to use health records and past antibiogram results to predict antimicrobial resistance in the ICU. Two new types of features are designed aiming to improve prediction performance. The first ones capturing information about resistant bacteria detected in past antibiograms for particular patients (p&*), and the second retrieving information of recently detected resistant germs in the ICU (r&*). To handle changes in data distribution over time caused by progressive mutation of bacteria, it is proposed to use an *incremental window* for the training set, and a test window with a 1-month fixed size such that training and test instances are temporarily as close as possible. The ML methods tested for classification are LR, KNN, RF, SVM and DS. DS is considered since it has been seen to perform well on the particular domain of this study. In addition to that, an ensemble of three feature weighting techniques (IG, CVD and UEB-1) is built to evaluate features relevance. Finally, feature selection and oversampling through Boderline-SMOTE are employed.

In the case of the analyzed data set, which contains concept drift and also suffers from data scarcity, experiments show that the proposed windowing scheme gets better results for the majority of antimicrobial families than standard train and test sliding windows found in literature.

The set of p&* features has proved to improve prediction in all experiments carried

out. In the first place, feature weighting has determined each of the p&* features as the most relevant feature to their respective antimicrobial family, except for POL family in which p&pol is the fourth most relevant feature. Also, the highest performance values in the study are obtained when p&* features are included in the data set, either just them together with initial features, or together with r&* and initial features. This is true to the point that the best result for CF4 prediction is achieved by just using p&cf4 feature as training. The utility of this type of features further reveals that knowing whether a patient was recently infected with a resistant bacterium is crucial to know if the patient is currently infected or not.

On the other hand, although r&* features exhibit a smaller improvement than p&* features, they have also shown a considerable utility for resistance prediction. In feature weighting, they are among the most relevant features of the data set. Some r&* features even get a higher importance than p&* features regarding the same target. Assessing their contribution to classification, it is observed that adding them, on their own, to the data set do not make much difference on prediction success. Nevertheless, when they are added together with p&* features, high performance results are provided, in some cases even higher than adding p&* features alone to the data set. The fact that r&* features contribute in some way to prediction means that information of recent resistant bacteria detected in patients of the ICU, contains a relatively high amount of information to predict bacteria resistance in other patients, which could indicate that bacteria are spreading among ICU patients.

Regarding the different ML methods, RF is the one that stands out over the rest. It has provided the best outcomes in the work for each of the antimicrobial families. The only exception is POL, in which the best performing method is SVM. Nonetheless, DS has shown high performance rates for some of the configurations tested in the experiments.

Feature selection has reduced complexity and enhanced classification in the majority of cases but, in addition, it has showed that feature weighting actually works in deciding which are the best features for prediction.

It is also interesting to acknowledge the usefulness of oversampling with Borderline-SMOTE which improves the prediction for half of the families by balancing the number of instances.

The final MDR classifier achieves an accuracy of a 75.0%, which can be seen as a quite good result considering the simplicity of the features employed, the limited amount of data used for training and the class imbalance observed in data set instances. With this it is concluded that successful outcomes are provided for simple resistance prediction, and furthermore it is observed that MDR prediction for *Pseudomonas aeruginosa* is possible by using demographic data of patients, information of their ICU admission, and historic antibiogram results.

5.1 Future work

As future work it is considered including further patient's details about their admission, such as the antibiotics they have been administered, whether they have required intubation, if they have needed mechanical ventilation, among others, which are indicators that have impact on the appearance of resistant bacteria [36]. SMOTE has considerably improved classification, but maybe a better performance could be achieved by using the method in a different manner. Class imbalance is mainly due to the fact that, for some time intervals there is a greater number of sensitive instances and for some other time intervals the majority of instances are resistant. In the way SMOTE has been used in this work, the instances generated for the minority class are generated for the whole training window, that is, balancing the totality of instances. In this case, it would be more interesting to balance separately different time intervals where resistance is similar. In that way, the training instances closest to test, which are the most important for prediction, would be class balanced. Finally, regarding the DS classifier, it may be interesting to test methods different from KNORA and different base classifiers, since DS methods has been seen to provide good results for this domain and it has shown a potential good performance in this study.

List of Figures

1 Exponential decay 16
2 Features correlation for `c&amg` 22
3 Evolution of antimicrobial resistance over time 24
4 Incremental training window scheme 26

5 Temporal evolution of `c&amg` prediction. 48
6 Temporal evolution of instances's classes for `c&amg` data set. 49
7 Temporal evolution of `c&qui` prediction. 49
8 Temporal evolution of instances's classes for `c&qui` data set. 50
9 Temporal evolution of MDR prediction. 51
10 Temporal evolution of instances's classes for MDR data set. 51

List of Tables

1 Features names and their descriptions. 14
2 Missing values proportion in p&* features. 19
3 Missing values proportion in r&* features. 20
4 Feature weighting for c&amg target 31
5 Feature weighting for c&car target 32
6 Feature weighting for c&cf4 target 33
7 Feature weighting for c&pap target 34
8 Feature weighting for c&pol target 35
9 Feature weighting for c&qui target 36
10 Incremental window evaluation 37
11 Results using the initial data set features 40
12 Hyperparameters for initial data-set features 40
13 Results using `p&*` features together with initial data set features . . . 41
14 Hyperparameters for `p&*` features and initial data-set features 41
15 Results using `r&*` features together with initial data set features . . . 42
16 Hyperparameters for `r&*` features and initial data-set features 43
17 Results using `p&*` and `r&*` features together with initial data set features 44
18 Hyperparameters for `p&*` and `r&*` features and initial data-set features 44
19 Summary of best results for each configuration of features 45
20 Results of feature selection 45
21 Results of oversampling with feature selection 46
22 Results of MDR predictor with optimal trained classifiers 48

Bibliography

[1] Demerec, M. Origin of bacterial resistance to antibiotics. *Journal of bacteriology* **56**, 63 (1948).

[2] Russell, A. Antibiotic and biocide resistance in bacteria: introduction. *Journal of applied microbiology* **92**, 1S–3S (2002).

[3] Revuelta-Zamorano, P. *et al.* Prediction of healthcare associated infections in an intensive care unit using machine learning and big data tools. In *XIV Mediterranean Conference on Medical and Biological Engineering and Computing 2016*, 840–845 (Springer, 2016).

[4] Hanberger, H. *et al.* Antibiotic susceptibility among aerobic gram-negative bacilli in intensive care units in 5 european countries. *Jama* **281**, 67–71 (1999).

[5] Brusselaers, N., Vogelaers, D. & Blot, S. The rising problem of antimicrobial resistance in the intensive care unit. *Annals of intensive care* **1**, 47 (2011).

[6] Vincent, J.-L. Nosocomial infections in adult intensive-care units. *The lancet* **361**, 2068–2077 (2003).

[7] Defez, C. *et al.* Risk factors for multidrug-resistant pseudomonas aeruginosa nosocomial infection. *Journal of Hospital Infection* **57**, 209–216 (2004).

[8] Weiner, L. M. *et al.* Antimicrobial-resistant pathogens associated with healthcare-associated infections: summary of data reported to the national healthcare safety network at the centers for disease control and prevention, 2011–2014. *infection control & hospital epidemiology* **37**, 1288–1301 (2016).

[9] Maragakis, L. L., Perencevich, E. N. & Cosgrove, S. E. Clinical and economic burden of antimicrobial resistance. *Expert review of anti-infective therapy* **6**, 751–763 (2008).

[10] Micek, S. T. *et al.* Pseudomonas aeruginosa bloodstream infection: importance of appropriate initial antimicrobial treatment. *Antimicrobial agents and chemotherapy* **49**, 1306–1311 (2005).

[11] Cantón, R. Lectura interpretada del antibiograma: una necesidad clínica. *Enfermedades Infecciosas y microbiología clínica* **28**, 375–385 (2010).

[12] Joshi, S. *et al.* Hospital antibiogram: a necessity. *Indian journal of medical microbiology* **28**, 277 (2010).

[13] Tsymbal, A., Pechenizkiy, M., Cunningham, P. & Puuronen, S. Handling local concept drift with dynamic integration of classifiers: Domain of antibiotic resistance in nosocomial infections. In *19th IEEE Symposium on Computer-Based Medical Systems (CBMS'06)*, 679–684 (IEEE, 2006).

[14] Timsit, J.-F., Harbarth, S. & Carlet, J. De-escalation as a potential way of reducing antibiotic use and antimicrobial resistance in icu (2014).

[15] Martínez-Agüero, S., Mora-Jiménez, I., Lérida-García, J., Álvarez Rodríguez, J. & Soguero-Ruiz, C. Machine learning techniques to identify antimicrobial resistance in the intensive care unit. *Entropy* **21**, 603 (2019).

[16] Hernàndez-Carnerero, À. *et al.* Modelling temporal relationships in pseudomonas aeruginosa antimicrobial resistance prediction in intensive care unit. In *Proc. of Singular Problems for Health Care (SP4HC) Workshop at the 24th European Conference on Artificial Intelligence (ECAI 2020)* (2020).

[17] Ellington, M. *et al.* The role of whole genome sequencing in antimicrobial susceptibility testing of bacteria: report from the eucast subcommittee. *Clinical microbiology and infection* **23**, 2–22 (2017).

[18] Pesesky, M. W. *et al.* Evaluation of machine learning and rules-based approaches for predicting antimicrobial resistance profiles in gram-negative bacilli from whole genome sequence data. *Frontiers in microbiology* **7**, 1887 (2016).

[19] Nguyen, M. *et al.* Using machine learning to predict antimicrobial mics and associated genomic features for nontyphoidal salmonella. *Journal of clinical microbiology* **57** (2019).

[20] Arango-Argoty, G. *et al.* Deeparg: a deep learning approach for predicting antibiotic resistance genes from metagenomic data. *Microbiome* **6**, 1–15 (2018).

[21] Tlachac, M. *et al.* Predicting future antibiotic susceptibility using regression-based methods on longitudinal massachusetts antibiogram data. In *HEALTHINF*, 103–114 (2018).

[22] Daelemans, W. & van den Bosch, A. Generalization performance of backpropagation learning on a syllabification task. In *Proceedings of the 3rd Twente Workshop on Language Technology*, 27–38 (Universiteit Twente, Enschede, 1992).

[23] Núñez, H., Sánchez-Marré, M. & Cortés, U. Improving similarity assessment with entropy-based local weighting. In *International Conference on Case-Based Reasoning*, 377–391 (Springer, 2003).

[24] Núñez, H. & Sànchez-Marrè, M. Instance-based learning techniques of unsupervised feature weighting do not perform so badly! In *ECAI*, vol. 16, 102 (2004).

[25] Widmer, G. & Kubat, M. Learning in the presence of concept drift and hidden contexts. *Machine learning* **23**, 69–101 (1996).

[26] Cruz, R. M., Hafemann, L. G., Sabourin, R. & Cavalcanti, G. D. Deslib: A dynamic ensemble selection library in python. *Journal of Machine Learning Research* **21**, 1–5 (2020).

[27] Ko, A. H., Sabourin, R. & Britto Jr, A. S. From dynamic classifier selection to dynamic ensemble selection. *Pattern recognition* **41**, 1718–1731 (2008).

[28] Chawla, N. V., Bowyer, K. W., Hall, L. O. & Kegelmeyer, W. P. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research* **16**, 321–357 (2002).

[29] Han, H., Wang, W.-Y. & Mao, B.-H. Borderline-smote: a new over-sampling method in imbalanced data sets learning. In *International conference on intelligent computing*, 878–887 (Springer, 2005).

[30] Zhang, S., Wu, X. & Zhu, M. Efficient missing data imputation for supervised learning. In *9th IEEE International Conference on Cognitive Informatics (ICCI'10)*, 672–679 (IEEE, 2010).

[31] Arauzo-Azofra, A., Aznarte, J. L. & Benítez, J. M. Empirical study of feature selection methods based on individual feature evaluation for classification problems. *Expert Systems with Applications* **38**, 8170–8177 (2011).

[32] Naylor, A., Hayes, P. & Darke, S. A prospective audit of complex wound and graft infections in great britain and ireland: the emergence of mrsa. *European Journal of Vascular and Endovascular Surgery* **21**, 289–294 (2001).

[33] Tornieporth, N. G., Roberts, R. B., John, J., Hafner, A. & Riley, L. W. Risk factors associated with vancomycin-resistant enterococcus faecium infection or colonization in 145 matched case patients and control patients. *Clinical Infectious Diseases* **23**, 767–772 (1996).

[34] Memmel, H., Kowal-Vern, A. & Latenser, B. A. Infections in diabetic burn patients. *Diabetes Care* **27**, 229–233 (2004).

[35] McGregor, J. C. *et al.* Utility of the chronic disease score and charlson comorbidity index as comorbidity measures for use in epidemiologic studies of antibiotic-resistant organisms. *American Journal of Epidemiology* **161**, 483–493 (2005).

[36] Rao, G. G. Risk factors for the spread of antibiotic-resistant bacteria. *Drugs* **55**, 323–330 (1998).