TÉCNICO LISBOA

Physician-friendly predictive model: Application to vasopressors administration and mechanical ventilation prediction

Filipe Santos Nobre da Costa

Thesis to obtain the Master of Science Degree in

Information Systems and Computer Engineering

Supervisors: Profª. Helena Isabel De Jesus Galhardas
Prof. João Carlos Serrenho Dias Pereira
Prof. Manuel João Caneira Monteiro da Fonseca

Examination Committee

Chairperson: Prof. José Carlos Alves Pereira Monteiro
Supervisor: Profª. Helena Isabel De Jesus Galhardas
Members of the Committee: Prof. João Paulo Baptista de Carvalho

October 2018

Acknowledgments

I would like to thank my parents, Elisabete and Pedro, and sister, Joana, for their encouragement over all these years. All I achieve is for them.

I would like to thank my girlfriend, Laura, for always being there for me. Your belief in me gives me strength to overcome everything.

I thank my supervisors, Helena, João and Manuel João, for their unconditional availability and for being so rigors with my work. They made me give my best.

Finally but not least I would like to thank my future business partner, Luís, for being supportive and available over all these years. Without him I would not be the person I am today.

To each and every one of you – Thank you.

Abstract

Nowadays, hospital units have large repositories of clinical data regarding the patient stays. These repositories have an enormous potential to be used to support physicians when they are making decisions. Despite its enormous potential, clinical data are complex (high dimensionality, temporal behaviour, etc.), posing challenges in its use in decision models. Nevertheless, there are several approaches proposing predictive models that explore the clinical data repositories. However, most of the approaches uses complex algorithms that are difficult to put in practice and are not focused on the interpretability of the predictions by the physicians. In this context, we propose two predictive models based on kNN for predicting vasopressors administration and mechanical ventilation within the next hours. The proposed predictive models use a small set of features and provide the clinical information of past patients considered for the predictions, making the predictions easier to interpret by physicians. Using only 5 features, for predicting vasopressors administration within 2 hours, the best model achieved an AUC of 0.927 and for predicting mechanical ventilation within two hours, the best model achieved an AUC of 0.906.

Keywords

Clinical decision support, Clinical data, Machine learning, Methodology

Resumo

Atualmente, as unidades hospitalares dispõem de grandes repositórios de dados clínicos referentes à estadia de pacientes. Estes repositórios têm um enorme potencial para serem utilizados como suporte aos médicos quando estes têm que tomar decisões. Apesar deste seu enorme potencial, os dados clínicos são complexos (elevada dimensionalidade, comportamento temporal, etc.) colocando desafios à sua utilização em modelos de decisão. Mesmo assim, existem várias abordagens que propõem modelos preditivos que exploram os repositórios com dados clínicos. Contudo, a maior parte destas abordagens utiliza algoritmos complexos que são difíceis de passar à prática e não estão focadas na interpretabilidade das previsões por parte dos médicos. Neste contexto, nós propomos dois modelos preditivos baseados no kNN para a previsão da administração de vasopressores e ventilação mecânica nas próximas horas. Os modelos preditivos propostos utilizam um conjunto de variáveis reduzido e fornecem com as previsões a informação clínica dos pacientes do passado considerados para as previsões, tornando as previsões facilmente interpretáveis pelos médicos. Utilizando apenas 5 variáveis, para a previsão da administração de vasopressores nas próximas duas horas, o melhor modelo obteve uma AUC de 0.927, para a previsão da ventilação mecânica nas próximas duas horas, o melhor modelo obteve uma AUC de 0.906.

Palavras Chave

Suporte à decisão médica; Dados clínicos; Aprendizagem automática; Metodologia;

Contents

List of Figures

List of Tables

1

Introduction

Contents

Over the last few years, we have witnessed a digital transformation in hospital units that resulted on the construction of huge clinical repositories containing data about past patients. Typically, these clinical repositories store data about the evolution of the values of certain clinical parameters/features (e.g., glucose, heart rate) and the treatments received during patients' stay in medical units. These large repositories constitute an enormous potential as a source of information during clinical decision situations (e.g., treatment prescription for a new patient) [7].

However, making decisions from the data generated in hospital units is challenging, due to the characteristics of the collected data, in particular:

1. *High dimensionality*: for each patient, clinical data repositories may store more than one hundred clinical features.

2. *Presence of outliers*: most of the data stored in clinical data repositories are introduced by humans which are error prone.

3. *Temporal data*: most of the features are composed by a sequence of measurements over time.

4. *Unequal length temporal features*: each patient, according to her problem, may require different types of monitoring. Therefore, different patients may have more measurements than others.

Despite these challenges, there are several works proposing predictive models to assist physicians when they are prescribing a treatment to a new patient [5, 6, 8, 9][1]. Fialho et al. [8] combine fuzzy modelling with feature selection to predict vasopressor administration within two hours in patients that required fluid resuscitation. They trained three different fuzzy models [11], one applied to the general population and the other two applied to patients with a specific disease, pneumonia or pancreatitis. The model applied to the general population achieved an Area Under the Curve (AUC)[2] of 0.79 $\pm$ 0.02, the model applied to patients with pneumonia achieved an AUC of 0.82 $\pm$ 0.02 and the model applied to patients with pancreatitis achieved an AUC of 0.83 $\pm$ 0.03. This study shows the potential of predicting vasopressors administration on the general population in intensive care units and also highlights the advantages of disease specific predictive models.

The work by Salgado et al. [5] can be seen as an extension of the work by Fialho et al. [8]. They propose an ensemble fuzzy model [5] to predict vasopressor administration on patients with pancreatitis and/or pneumonia, obtaining an AUC of 0.85 $\pm$ 0.01.

Wu et al. [9] focus on three important tasks: (i) imminent vasopressor need (i.e., requiring vasopressor within the next two hours); (ii) short term vasopressor need (i.e., not requiring vasopressors for the

[1] We focus our analysis on these works, because all of them used the public clinical data repository MIMIC III [10], containing health related data about 40,000 patients who stayed in critical care units of the Beth Israel Deaconess Center between 2001 and 2012. We propose to use the same data repository to enable a fair comparison.

[2] AUC measures the entire two-dimensional area underneath the entire ROC curve (plot of true positive rate versus false negative rate).

next 4 hours but requiring it in the following 2 hours); and (iii) wean readiness (i.e., successful wean is when a patient does not require vasopressors again within 4 hours). In tasks (i) and (ii), the authors make hourly predictions until the first vasopressor administration or the end of stay. For all tasks, they use the latent states from a Switching-State Autoregressive Model (SSAM) [12] combined with raw data (physiological data + static admission data) as features, achieving respectively, for each task, AUCs of: (i) 0.92 ± 0.0016, (ii) 0.88 ± 0.0061, and (iii) 0.71 ± 0.005.

Later, M. Ghassemi et al. [6] tried to extend the work done by Wu et al. to the prediction of five ICU treatments: mechanical ventilation, vasopressor administration, and three blood transfusions. Again, the authors applied the latent states from a SSAM with static data of patients (e.g. gender), obtaining an AUC of 0.82 for patients that require vasopressors within the next hour and an AUC of 0.68 for patients that require mechanical ventilation within one hour.

In the previously described works we found the following limitations:

1. The proposed models use a large set of features (see Table 1.1), which reduces the interpretability of the models. In fact, it is difficult to interpret a prediction if we need to analyze a large set of features;

2. Predictions are not labeled with intuitive information explaining the reasons behind them. Without this additional information, it is very difficult for physicians to understand and validate the predictions;

3. Some of the proposed models are generated by complex algorithms that require time to understand and put in practice.

Table 1.1: Number of features used by the predictive models proposed in the state of the art.

Paper	# features
Fialho et al. [8]	10
Salgado et al. [5]	24
Wu et al. [9]	19
M. Ghassemi et al. [6]	29

1.1 Goals

The main goal of this work is to propose and evaluate two predictive models: one for predicting vasopressors administration and the other for predicting mechanical ventilation, applied to the MIMIC-III repository. We aim at obtaining predictive results similar or better than the results of the state of the art [5, 6, 8, 9] but with the advantage that the predictions can be easily interpreted by physicians. In particular, the proposed models should overcome the limitations of the state of the art in the following way:

1. Use a small set of features to enable physicians to easily interpret the predictions;
2. Provide additional information with the predictions so that physicians can take a justified decision when accepting or rejecting the predictions;
3. Follow a well defined methodology in the construction of the predictive models, composed by simple steps and methods.

1.2 Contributions

The main contributions of this thesis are as follows:

1. Two predictive models based on k-Nearest Neighbor (kNN), for the prediction of vasopressors administration and mechanical ventilation. The proposed models only use 5 features and provide additional clinical information of past patients used for the prediction;
2. Experimental validation of the proposed models in a large and public clinical repository;
3. A methodology for the construction of medical predictive models. This methodology encompasses a set of well defined steps that suggest actions to use the data stored in the clinical repositories in order to create a predictive model capable of assisting physicians when they making decisions. The methodology is composed by four step: (i) data pre-processing, (ii) selection of the patient representation, (iii) feature ranking and (iv) construction of the predictive model.

The work presented in this thesis was the basis for a research paper accepted and presented at the 10th INForum - *Simpósio de Informática* (INForum 2018) [13].

1.3 Document Outline

This document is organized as follows: Section 2 describes the dataset used and reviews the traditional clustering and classification algorithms, and their applicability to time series, and, in particularly, to clinical data considering the goals of the work. Section 3 describes the steps of the proposed methodology for the construction of medical predictive models. Section 4 describes the construction and validation of a model for the prediction of vasopressors administration. Section 5 describes the construction and validation of a model for the prediction of mechanical ventilation. Finally, Section 6 presents the conclusion, with a summary of the developed work and the future lines of work.

2

Related Work

Contents

In this chapter, we start by studying clustering and classification algorithms in order to find the most suitable methods to incorporate in the proposed methodology. In particular, we describe: (i) traditional clustering algorithms (Section 2.1), biclustering algorithms (Section 2.2) and their application on temporal data (Section 2.3); and (ii) the most used classification algorithms (Section 2.4). Then, we compare the clustering and classification algorithms considering the goals of this thesis (Section 2.5). At the end of this chapter, we describe the clinical repository used in this work (Section 2.6).

2.1 Clustering Algorithms

Clustering is a well known technique used to group unlabeled data so that the similarity within-groups is maximized and the similarity between-groups is minimized. Clustering algorithms can be classified into five categories [14]: (i) partitioning methods; (ii) hierarchical methods; (iii) density-based methods; (iv) grid-based methods; and (v) model-based methods.

2.1.1 Partitioning Methods

Partitioning methods construct K partitions of the data, where each partition contains at least one instance. The partition can be crisp or fuzzy. If each instance belongs to only one cluster, then the partition is crisp, if one instance can belong to more than one cluster, then the partition is fuzzy. K-means [15] is an algorithm that uses crisp partitions and Fuzzy c-means [16, 17] is an algorithm that uses fuzzy partitions.

K-means algorithm is probably one of the most famous clustering algorithms. It is possible to see in the scientific literature the existence of an enormous amount of variations of this algorithm [17–19]. K-means can be seen as a minimization problem. Let X be the set of all d-dimensional points and C be the set of K disjoint clusters, where $C = \{c_k, k = 1, ..., K\}$ and μ_k are the center of a cluster c_k (equation 2.1). The goal of K-means is to minimize an objective function, which is typically the sum of the squared errors between μ_k and c_k over all K clusters (equation 2.2).

$$\mu_k = \frac{1}{\#c_k} \sum_{x \in c_k} x \tag{2.1}$$

$$\sum_{k=1}^{K} \sum_{x \in c_k} \|x - \mu_k\|^2 \tag{2.2}$$

In a high level detail the main steps of K-means are:

1. Initialize (e.g., randomly) the center of each cluster (μ_k).

2. Decide the cluster membership of all instances, by assigning them to the nearest cluster center.

3. Recalculate the new cluster centers (μ_k).

4. If all clusters memberships are equal to the memberships from the previous iteration or the difference is smaller than a threshold, return the membership of all instances; Otherwise, go to step 2.

Obtaining a K-means solution is an NP-hard problem. It acts like a greedy algorithm, so there is no guarantee about the convergence, but typically converges to a local minimum. Another important aspect of the algorithm is the choice of three required user-specified parameters: (i) the number of clusters K; (ii) the cluster initialization; and (iii) the distance metric. Regarding the number of clusters K, there is no law, although in the literature, it is possible to find different heuristics for choosing a better K [20]. These heuristics should be combined with the evaluation of domain experts, and the most meaningful value of K should be selected. The cluster initialization also has an important role. According to some studies [21] cluster initialization aligned with some data properties makes it possible to converge to a global minimum with a large probability. Moreover, another way to fight the local minimum convergence is to run the algorithm multiple times with different cluster initializations and pick the best one. Finally, concerning the distance metric, K-means is typically used with Euclidean Distance [1], resulting always in spherical clusters. However, this is not a strict rule. Depending on the domain specificities, some distance measures can be more accurate than others. For example, the Itakura Saito distance has been successfully used in speech processing [22]. Despite some disadvantages, K-means can have linear time complexity [23], and its time efficiency makes it commonly used.

Fuzzy c-means proposed by Dunn [16] and generalized by Bezdek [17] is an extension of K-means where each instance can belong to more than one cluster with different membership values. So the previous equation 2.2 is changed to :

$$\sum_{k=1}^{K} \sum_{x \in c_k} (m_{xk})^j \|x - \mu_k\|^2 \tag{2.3}$$

where m_{xk} is a real number that represents the membership of a data point x to a cluster k ($m_{xk} \in [0, 1]$) and j is any real number greater than one. To actualize these two variables m_{xk} and μ_k, we need to derive the objective function with respect to both variables to get the following equations:

$$\mu_k = \frac{\sum_{x \in c_k} (m_{xk})^j x}{\sum_{x \in c_k} (m_{xk})^j} \tag{2.4}$$

$$m_{xk} = \frac{(1/\|x - \mu_k\|^2)^{\frac{1}{m-1}}}{\sum_{k=1}^{K} (1/\|x - \mu_k\|^2)^{\frac{1}{m-1}}} \tag{2.5}$$

The main steps of Fuzzy c-means are:

1. Initialize (e.g., randomly) the memberships of each data point (m_{xk}).
2. Calculate the clusters centers (μ_k) using equation (3).
3. Update the memberships (m_{xk}) using equation (4).
4. If all clusters memberships are equal to the memberships from previous iteration or the difference is smaller than a threshold, return the membership of all instances; Otherwise, go to step 2.

Fuzzy c-means has several disadvantages [24], for instance, it has a slower convergence speed than K-meas and it is highly sensitive to the initialization step. The major advantage is allowing different memberships.

2.1.2 Hierarchical Methods

Hierarchical clustering algorithms construct clusters by recursive data merging (agglomerative strategy) or data splitting (divisive strategy). The recursive merging or splitting process is represented as a tree, usually called dendrogram. Agglomerative hierarchical clustering starts by considering each instance as a cluster and then merges these atomic clusters into larger and larger clusters until a stopping criteria is met. That can be, for instance, when we have a single cluster containing all instances. On the opposite side, the divisive approach starts with all instances in a single cluster and then splits this cluster into smaller and smaller clusters until a stopping criteria is met. The most decisive step is when the algorithm needs to select the best next cluster(s) to split or merge [25] .

In the agglomerative clustering approach there are mainly three methods to evaluate if two clusters are closer or not :

- *Single link method* - The distance between two clusters A and B is given by the minimum distance of two instances, where one instance belongs to A and the other belongs to B.
- *Complete link method* - The distance between two clusters A and B is given by the maximum distance of two instances, where one instance belongs to A and the other belongs to B.
- *Average link method* - The average of the distance between all instances of pairs of clusters.

Two clusters are merged based on a minimum distance criteria. Depending on the method chosen, the result will be different. The complete link usually produces compact clusters. By contrast, single link tends to produce elongated clusters. In the literature, it is possible to find often that the complete link method produces more useful hierarchies than the single link algorithm [26].

In divisive clustering, as it was said before, the most important aspect is how to choose the next cluster to split. Two possible approaches are [25] :

- *Size-priority cluster split* - Select the cluster with the largest size to split, formally choose the cluster p, so that $p = argmin_k(1/n_k)$, where n_k is the number of instances of a cluster k.

- *Average similarity* - The idea is to choose the smallest cluster p with the smallest average similarity, $p = argmin_k(s_{kk}/n_k^2)$, where n_k is the number of instances of a cluster k and s_{kk} is the self-similarity of a cluster C_k, $s(C_k, C_k)$. In other words, the sum of pairwise similarities within C_k, $\sum_{i \in C_k} \sum_{j \in C_k} w_{ij}$, where w_{ij} is the similarity between i,j.

One important aspect of agglomerative and divisive methods is that when two clusters are merged or split, it is not possible to reverse this decision anymore. There are evidences [27] that divisive algorithms are more accurate than agglomerative algorithms, because the former benefit from complete information about the global distribution, while the latter make decisions based only on local patterns, performing more mistakes in initial stages that cannot be undone.

Bisecting K-means is a good example of an efficient divisive clustering algorithm, because it explores the time efficiency of K-means and the quality of divisive clustering algorithms [27]. It starts with a single cluster and works in the following manner:

1. Pick a cluster to split.

2. Use K-means to find two sub-clusters, basically use K-means with K=2.

3. Repeat step 2, for a fixed number of times and take the split that produces the clustering with the highest overall similarity.

4. Repeat steps 1, 2 and 3 until the desired number of clusters is reached.

One of the major advantages of hierarchical methods is the generation of visual dendrograms, which can assist the end-user during the clustering process. Another important advantage is the deterministic behavior, compared to nondeterministic methods like K-means [1].

The major disadvantages are the impossibility of undoing merging or splitting decisions and the quadratic complexity time, which is not desirable especially for large datasets.

2.1.3 Density-Based Methods

Density-based clustering algorithms construct clusters by grouping a set of instances spread in a continuous region of high density of data. Each cluster is separated from other clusters by continuous regions of low density of data that are not assigned to any cluster [28]. These dense connected areas can have arbitrary shapes, so there is no assumptions about the clustering shapes, as we have in K-means, where clusters can only be spherical. Furthermore, density-based methods do not make any assumption about the number of clusters that exists neither their distribution.

DBSCAN (Density Based Spatial Clustering of Applications with Noise) [29] is a density-based clustering algorithm that was designed to find clusters and noise in spatial databases. The key idea is that, for each point of a cluster, the neighborhood of a given radius (ϵ) has to contain at least a minimum number of points (MinPts), i.e., the density in the neighborhood has to exceed some threshold. All points that respect this condition are called *core points*. First, it is important to clarify how to quantify this density concept. The *Eps-neighborhood* of a point p ($N_{Eps}(p)$) is constituted by all points q, for which d(p,q) $\leq$ Eps , where d(p,q) is any possible distance function (e.g. Euclidean distance, Manhattan distance). However, in general, the Eps-neighborhood of points in the border of the cluster (*border points*), contains significantly less points than the Eps-neighborhood of the *core points*. So, we need to add another rule in order to include these *border points* more accurately. A point p is *directly density-reachable* from a point q with respect to Eps and MinPts, if $p \in N_{Eps}(q)$ and $|N_{Eps}(q)| \geq MinPts$. A point p is *density-reachable* from a point q with respect to Eps and MinPts if there is a chain of points $p_1, ..., p_n, p_1 = q, p_n = p$ such that p_{i+1} is *directly density-reachable* from p_i. With this concept of *density-reachable*, we still can not define a proper cluster, because there is the possibility of two *border points* of the same cluster C not be *density-reachable* from each other, the *core point* condition may not hold for both of them. However, there must be a *core point* in C from which both *border points* of C are *density-reachable*, and this corresponds to the concept of *density connectivity*, i.e. a point p is *density-connected* to a point q with respect to Eps and MinPts, if there is a point o such that both p and q are *density-reachable* from o with respect to Eps and MinPts.

We are now able to define a density-based cluster as a maximal set of density-connected points. Noise is the set of points that do not belong to any cluster.

To find a cluster DBSCAN starts with an arbitrary point p and finds all density reachable points with respect to Eps and MinPts. If p is a core point, this process yields to a cluster. If p is a border point, no points are density-reachable from p and DBSCAN visits the next point of the database. The algorithm finishes when all points have been assigned to a cluster or to noise. The overall complexity is O(NlogN), where N is the number of points of the database.

One potential problem of DBSCAN is the definition of the two required parameters Eps and MinPts, but we can find heuristics that can help to minimize this problem [29]. To overcome this problem the OPTICS algorithm [30] computes an augmented cluster ordering for automatic and interactive cluster analysis. The ordering contains information that is equivalent to density-based clustering obtained from a wide range of parameter settings [31].

2.1.4 Grid-Based Methods

Grid-based methods [32–34] partition the data space into a finite number of cells to form a grid structure and then form clusters based on the principle that regions that are more dense than their

surroundings correspond to clusters. Most grid based methods may also be considered as density based methods. The advantage of this family of clustering algorithms is a significant reduction in time complexity, because clustering is applied on cells rather than on data points. In most applications, the number of cells is significantly smaller than the number of points.

Grid-based methods typically involve the following steps [35, 36]:

1. Create the grid structure, i.e., partition the data space into a finite number of cells.
2. Calculate the cell density for each cell.
3. Sort the cells according to their densities.
4. Identify cluster centers.
5. Traverse neighbor cells.

A classical approach is the STatistical INformation Grid (STING) algorithm [33] to cluster spatial databases. This method divides the spatial area into rectangular cells, which are represented by a hierarchical structure. In other words, the spatial data is divided into levels, where the level 1 is the root cell of the hierarchy and represents all spatial area, its cell children's are at level two, and so forth until a specific number of levels is reached. A cell at level i corresponds to the union of all areas of its children at level i + 1.

STING maintains summary statistics for each cell in the hierarchical tree. The algorithm adopts a top-down approach for clustering and querying. For each cell in the current layer considered, a confidence interval is computed reflecting the cell's relevance to the given query. If the cell is relevant to the query, the process continues to the next lower level until the bottom layer is reached, or the answer to the query is met. Since statistics are saved, STING can be seen as a query-independent approach, where the time complexity for clustering is linear with the K leaves of the tree.

The disadvantages of grid-based clustering algorithms are: (i) how to choose the grid size or density thresholds and (ii) the curse of high dimensionality. The former problem can be minimized by using adaptive grids that automatically determine the size of grids based on the data distribution and does not require the user to specify any parameters like the grid size or the density thresholds [37]. The latter problem can be minimized using the algorithm OptiGrid [38], which selects relevant attributes by optimizing the density function over the data space.

2.1.5 Model-Based Methods

Model-based methods [39] attempt to fit the observed data to some mathematical model using a probabilistic approach. These methods are based on the assumption that data are generated by a

mixture of probabilistic distributions, such as Gaussian or Bernoulli distributions. So, the clustering problem can be converted to a parameter estimation of the K components (clusters) distributions that represent the data.

Formally, a distribution *f* is a mixture of K components distributions $f_1, f_2, ..., f_k$, if:

$$f(x) = \sum_{k=1}^{K} \pi_k f_k(x) \tag{2.6}$$

Where $\pi_1, \pi_2, ..., \pi_k$ are the mixing weights, satisfying the following conditions: 1) $\pi_k > 0$ and 2) $\sum_{k=1}^{K} \pi_k = 1$. Typically, f_k are all from the same parametric family, with different parameters (e.g. they can be all Gaussians with different mean values). So, we rewrite equation 2.6 to reflect the parameters of the K cluster (θ_k):

$$f(x) = \sum_{k=1}^{K} \pi_k f_k(x|\theta_k) \tag{2.7}$$

Now, we need to estimate the parameters of the distributions, and we can use the maximum likelihood technique [40]. The likelihood is the probability of the data being generated from the model with the parameters θ ($P(Dataset|\theta)$) and assuming that the data are drawn independently from the distribution:

$$P(Dataset|\theta) = \prod_{n=1}^{N} \sum_{k=1}^{K} \pi_k f(x_n|\theta_k) \tag{2.8}$$

The maximum likelihood can now be written as follows:

$$\theta^* = argmax_\theta\{P(Dataset|\theta)\} = argmax_\theta\{lnP(Dataset|\theta)\} \tag{2.9}$$

The Expectation-Maximization (EM) [40] is an approach to find the maximum likelihood estimates for the parameters θ for models with latent variables (not directly observed) where the Gaussian mixture model is an example. The EM works in the following manner:

1. Starts with guesses (e.g. randomly) about the cluster distributions $\theta_1, \theta_2, .., \theta_k$ and the mixing weights $\pi_1, \pi_2, .., \pi_k$.

2. Using the current guesses, calculates the probability of each point x_j being generated by each component (cluster) i, $p_{ij} = \frac{\pi_j f(x_i|\theta_j)}{\sum_{k=1}^{K} \pi_k f(x_i|\theta_k)}$ (*E-step*).

3. Maximizes the weighted likelihood to get new parameter estimates, using the current weights (*M-step*).

4. Go to step 2, if there are changes on the parameter estimates. Otherwise, return the final parameter estimates and cluster probabilities.

Despite the popularity of EM due to its simple implementation, and light analytical preparatory work [41], it has several well known limitations. The EM algorithm converges to a local minimum and its solutions are highly dependent of the initialization. The possible solutions to minimize these problems are [42]: (i) multiple random starts and choose the final estimate with the highest likelihood; and (ii) initialize using clustering algorithms. Moreover, like K-means, EM assumes that the number of clusters are known for modeling the distributions. However, in many applications, the number of clusters are unknown, so we need to try it with different number of clusters and pick the best model.

2.2 Biclustering

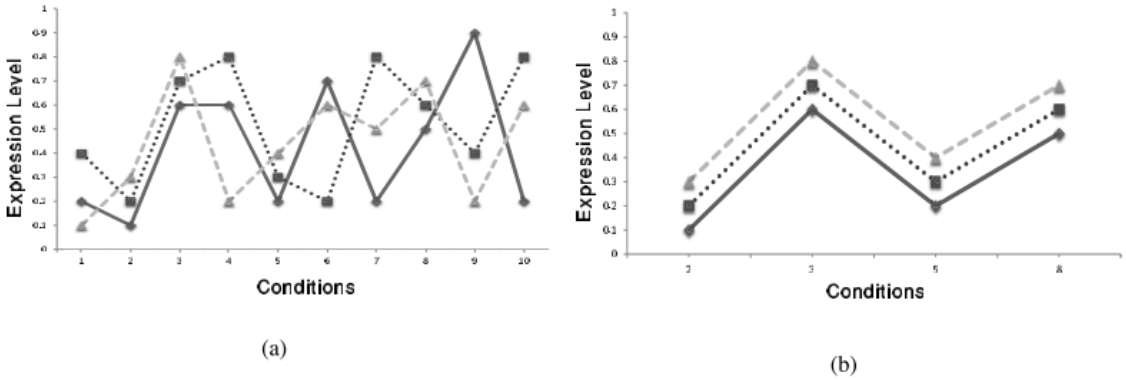


Figure 2.1: An illustration of biclustering. The expression levels of three genes over 10 different biological conditions are shown. (a) The genes are uncorrelated when all of the 10 conditions are considered. (b) The genes are strongly correlated in a subset of the conditions 2, 3, 5, 8. Example from [1].

In gene expression data, researchers are interested in study the level of genes within a number of different experimental samples (conditions). If we want to understand the relations between genes, the traditional clustering algorithms assume that related genes must have similar expression profiles across all the conditions. In many biological experiments, this strong assumption may lead to inconclusive results, because some subsets of genes can be coregulated and coexpressed only under a subset of conditions, but behave almost independently under other conditions. This inability to find local patterns instead of just global patterns was the catalyst to move beyond the clustering paradigm, and start a new one, called biclustering [43].

The objective of biclustering is to simultaneously cluster both rows and columns in a given matrix. Biclustering algorithms aim to discover local patterns that cannot be identified by the traditional one-way clustering algorithms [1]. To express the powerful of biclustering, lets analyze Figure 2.1. In Figure 2.1, the expression levels of three genes over 10 conditions are shown. Considering all of the ten samples, it is evident that there is no strong correlation between the three genes (Figure 2.1 (a)). However, there

Table 2.1: Examples of Biclustering Algorithms

Algorithm	Type	Structure
Block Clustering [47]	Constant	Nonoverlapping biclusters with tree structure
FLOC [48]	Coherent Values	Arbitrarily positioned overlapping biclusters
PRMs [49]	Constant Columns	Exclusive row and column biclusterss

is a strong correlation between the three genes in a subset of the conditions: 2, 3, 5, 8 (Figure 2.1 (b)). With traditional clustering algorithms we are not able to find correlation between these three genes.

There are several challenges that arise while we are searching for biclusters: (i) finding all the significant biclusters has been proven to be an NP-hard problem [44]; (ii) allowing positive and negative correlations in the same bicluster [45]; and (iii) allowing overlapping between clusters [46]. Concerning the types of biclusterings that the algorithms are able to find, there are four major classes [43]:

1. Biclusters with constant values.
2. Biclusters with constant values on rows or columns.
3. Biclusters with coherent values ($a_{ij} = \mu + \alpha_i + \beta j$ or $a_{ij} = \mu + \alpha_i * \beta j$, where a_{ij} is a subset of rows and columns, μ is a typical value within the bicluster, α_i is the adjustment for row and βj is the adjustment for column j).
4. Biclusters with coherent evolutions (biclusters with coherent behaviors regardless the exact numeric values in the data matrix. This behaviour can be expressed on the entire bicluster or on the rows or columns of the bicluster).

Another important aspect about the biclustering algorithms is how they deal with variations in the size and positioning of the biclusters. They may be able to find only one bicluster or K biclusters, where K is usually defined apriori. When the algorithm assumes the existence of K biclusters, several structures can be obtained, for instance nonoverlapping biclusters with checkerboard structure or arbitrarily positioned overlapping biclusters.

Table 2.1 illustrates some examples of biclustering algorithms that should be chosen according to the specificities of the problem. A detailed list can be found in this survey about biclustering [43].

2.3 Time Series Clustering Analysis

Time series represents ordered value measurements usually at regular temporal intervals. Formally, a *time series* $X = \{x_1, ..., x_n\}$ for $T = t_1, ..., t_n$ is a discrete function with value x_1 for t_1 and so on. *Time series* is a common form of data encountered in many different scenarios such as the stock markets, sensor data, fault monitoring, machine state monitoring, environmental applications, or medical data [1].

Clustering *time series* depends both on the type of data and the purpose of the study. Conventional cluster algorithms, summarized in Section 2.1, have very interesting results in many different domains, however they were typically designed to work with static data and not with data that varies over time. To overcome this problem we have three possible strategies: (i) change the conventional cluster algorithms to work with time series, often by changing the similarity measure in order to be possible to compare *time series* (raw-data based approach or shape based approach); (ii) convert the raw *time series* data to a feature vector of lower dimension; and (iii) convert the raw *time series* data to a number of model parameters, and then apply the conventional cluster algorithms. The last two strategies are called feature-based approach and model based-approach [31], respectively.

Clustering *time series* is not the same as clustering static data, so in the next subsections we are going to analyze: (i) the importance of choosing the similarity function; and (ii) the advantages and disadvantages of the traditional clustering algorithms and biclustering when applied to *time series*.

2.3.1 Similarity/Distance Functions

We need to define similarity/distance functions in order to compare *time series*. In the literature, there are many proposed distance functions, where each of them is appropriate for different applications and has specific advantages and disadvantages [31]. One distance function widely used is the Dynamic Time Warping (DTW) [50]. This function is particularly interesting when we have *time series* of unequal length. Given two one-dimensional *time series*, $Q = q_1, q_2, ..., q_i, ..., q_n$ and $R = r_1, r_2, ..., r_j, ..., r_m$, DTW aligns the two series so that their difference is minimized.

DTW computes the warping path W = $w_1, w_2, ..., w_K$ with $max(m, n) \leq K \leq m + n - 1$ for two *time series* q and r with lengths m and n, respectively. To compute this path, first we need to produce a matrix N x M, where the entries (i,j) represent the cumulative distance, $dcum(i, j) = d(q_i, r_j) + min[dcum(i - 1, j - 1), dcum(i - 1, j), dcum(i, j - 1)]$, and d is any distance function, usually euclidean. The warping path is the set of previous matrix elements that satisfy these three constraints: boundary condition, continuity, and monotonicity. The boundary condition constraint requires that the warping path starts and finishes in diagonally opposite corner cells of the matrix. The continuity constraint restricts the allowable steps to adjacent cells. The monotonicity constraint forces the points in the warping path to be monotonically spaced in time. The warping path besides satisfying the previous tree constraints represents the minimum distance possible between the two *time series*.

Euclidean distance is not able to deal with temporal drifting, neither unequal length *time series*. To deal with these problems DTW is a good choice.

Another distance function widely used is the Longest Common Sub Sequence (LCSS) [51]. The basic idea is to match two sequences by allowing them to stretch, without rearranging the sequence of the elements but allowing some elements to be unmatched. This measure can also deal with unequal

length *time series* and temporal drifting. Moreover, LCSS is more robust to outliers than Euclidean distance or DTW.

2.3.2 Partitioning Methods

One of the most important advantages of partitioning methods is their reduced time complexity. Since *time series* are high volume data, these methods are very suitable for clustering *time series* and have been used in many different works [52]. If *time series* have the same length and are aligned, we can use the euclidean distance as the distance function. If we have time series with different lengths, euclidean distance cannot be applied. One possible solution is to use DTW as our distance function. However, DTW averaging produces sequences of equal or greater length than the original ones, thus decreases the clustering system accuracy, because the new cluster centers do not preserve the characteristics of the cluster objects [1]. In [53], the authors proposed a shape-based K-means clustering technique that uses DTW as distance measure and improves the time complexity of DTW averaging.

2.3.3 Hierarchical Methods

The advantage of hierarchical methods when applied to *time series* is their visualization power, which makes it an approach to be used for *time series* clustering to a great extent. Another very important advantage is the fact that hierarchical clustering does not require a predefined number of cluster. Since *time series* typically represent real world problems, it is very hard to know *a priori* the number of clusters. Moreover, it is possible to cluster unequal length *time series*, if an appropriate elastic distance measure is used, such as Dynamic Time Warping (DTW). On the other hand, traditional hierarchical clustering algorithms are not capable of dealing effectively with large *time series* data, because they may have a quadratic computational complexity, making it suitable only for small datasets.

2.3.4 Density-Based Methods

Traditional density-based methods like DBSCAN cannot be effectively used in high volume data such as *time series*. One solution to this problem is for instance the model proposed by Chandrakala and Chandra [54]. However, in the literature, density-based clustering algorithms have not been used broadly for *time series* data clustering, because of its rather high implementation complexity [55].

2.3.5 Grid-Based Methods

Grid-based methods like density-based methods suffer from the curse of high dimensionality, and these methods require, as input, the grid size and density thresholds. All these disadvantages dis-

courages its application on *time series* data, and in the literature there is no work in the application of grid-based approaches for clustering *time series*, as far as we known.

2.3.6 Model-Based Methods

Model-based methods have been applied successfully to *time series*. However, these methods have two drawbacks: (i) can integrate background knowledge which can lead to inaccurate results; and (ii) have a slow processing time on large data sets [56].

2.3.7 Biclustering

Most biclustering formulations are NP-hard and thus heuristic approaches are often used. However optimal solutions are not guaranteed [43].

Most existing biclustering algorithms are not able to find biclusters with contiguous columns, and since there is an important internal sequential relationship in time-series data, these methods are not suitable for the analysis of *time series* data [57]. In the last decade, new biclustering algorithms have been created in order to explore the potential biological information of contiguous time points and find the co-expressed relationship among genes, k-CCC algorithm [57] and e-CCC algorithm [58] are two examples of successful biclustering algorithms applied to *time series*.

2.4 Supervised Learning - Classification

The goal of classification methods is to train a classifier from labeled data to predict the labels of unknown data. In this section, we describe the most well known classification methods: (i) instance-based classifiers; (ii) statistical classifiers; (iii) decision trees; (iv) support vector machines; (v) neural networks; and (v) ensemble methods.

2.4.1 Instance-Based Classifiers

Instance-based classifiers are lazy-learning algorithms, which means that they delay the generalization process until the classification is performed. The simplest method is k-Nearest Neighbour (kNN) [59].

k-Nearest Neighbour is based on the principle that instances that have similar properties are close to each other. So, in order to classify a new instance I, kNN finds the k closest instances based on a distance function (e.g., euclidean distance) and then assigns the most frequent label, between the k instances, to the new instance I. If k is an even number, it is possible to have draws, and in these cases the resulting class can be obtained randomly.

For more accurate results, several algorithms assign more importance to instances closer to the new instance than to instances far away. In other words, each point votes proportionally to its distance to the new instance [60].

Despite kNN being a very simple method, it has shown high accuracy when applied to different domains. However, there are some reservations about kNN [2, 61]: (i) large storage requirements (all dataset is required to find the k nearest instances); (ii) sensitive to the choice of the distance function that is used to compare instances; (iii) to choose the best k, it is required computationally-expensive techniques like cross validation; and (iv) sensitive to noise; if k is small, the noisy instances can win the majority of the votes;

2.4.2 Statistical Classifiers

Statistical classifiers are characterized by having an explicit underlying probability model. Instead of just giving the classification of an instance, they provide the probability of an instance belonging to each class. Naive Bayes classifier [62] is the most famous statistical learning algorithm.

The bayesian approach classifies new instances with the most probable class given the instances attributes values $a_1, a_2, ..., a_n$. Formally:

$$Classification = \underset{c_k \in C}{argmax} \quad P(c_k|a_1, a_2, ..., a_n) \tag{2.10}$$

Where C is all possible classifications. Using the Bayes theorem equation (2.11) we can rewrite the previous equation 2.10 :

$$P(h|D) = \frac{P(D|h) * P(h)}{P(D)} \tag{2.11}$$

$$\underset{c_k \in C}{argmax} \quad \frac{P(a_1, a_2, ..., a_n|c_k) * P(c_k)}{P(a_1, a_2, ..., a_n)} \tag{2.12}$$

Since $P(a_1, a_2, ..., a_n)$ is constant, we can dropped it out from the equation:

$$\underset{c_k \in C}{argmax} \quad P(a_1, a_2, ..., a_n|c_k) * P(c_k) \tag{2.13}$$

We need to estimate the two terms of equation 2.13. The last term $P(c_k)$ is quite easy to estimate, we just need to count the frequency of each class c_k occurring in the training data. However, estimating the $P(a_1, a_2, ..., a_n|c_k)$ is not feasible unless we have a very large set of training data [63]. If we have a small set of training data, we will not have reliable estimates. Moreover, suppose that the instance I with values $a_1, a_2, ..., a_n$ is unknown, so the probability of $P(I|c_k)$ will be zero for all classes, in consequence we are not able to classify this instance without using other method.

To solve this problem of estimating $P(a_1, a_2, ..., a_n|c_k)$, naive bayes classifier is based on the as-

sumption that the attributes values $a_1, a_2, ..., a_n$ are conditionally independent given the class c_k. In other words, the assumption is that given the class of the instance, the probability of observing the conjunction $a_1, a_2, ..., a_n$, is just the product of the probabilities for the individual attributes:

$$P(a_1, a_2, ..., a_n|c_k) = \prod_{i=1}^{n} P(a_i|c_k) \tag{2.14}$$

We can now rewrite the equation 2.13 with the naïve bayes classifier approach:

$$NaiveBayesClassification = \underset{c_k \in C}{argmax} \quad \prod_{i=1}^{n} P(a_i|c_k) * P(c_k) \tag{2.15}$$

The advantages of naive bayes classifier are: (i) short computational time for training and classifying [2]; (ii) not sensitive to irrelevant features; (iii) can make probabilistic predictions.

The disadvantages of naive bayes classifier are: (i) very strong assumption behind, which can lead to inaccurate results; (ii) Continuous attributes require special treatment; and (iii) usually requires big dataset in order to have reliable estimations.

2.4.3 Decision Trees

Decision trees are among the most popular machine learning algorithms and have been successfully applied to a broad range of tasks from learning to diagnose medical cases to learning to assess credit risk of loan applicants [63].

Decision trees classify instances using a tree structure. This tree structure is composed by nodes and branches. Each node specifies a test to some attribute of the instance, and each branch corresponds to one of the possible values for this attribute. To classify a new instance, we just need to start at the root node of the tree and move down in the tree until a leaf node is reached. The leaf node reached represents the classification of the new instance.

Most decision trees cannot perform well with problems that require diagonal partitioning, because the partition of space is orthogonal to the axis of one variable and parallel to all the other axis. Therefore, the resulting regions after partitioning are all hyper rectangles [64].

One important step in any decision tree algorithm, is the choice of the attribute to be tested. There are different methods for finding the feature that best divides the training data, such as information gain [65] or gini index [66].

The most well known algorithm for building decision trees is C4.5 [67], which is an extension of ID3 algorithm [68]. The idea behind ID3 algorithm is to recursively split the training dataset according to the

attribute with the highest information gain, where the information gain of an attribute S is :

$$InformationGain(S) = Entropy(Parent) - \sum_{a \in A} \frac{\#\{x \in Dataset : x.S = a\}}{\#\{x \in Dataset\}} * \tag{2.16}$$

$$* Entropy(x \in Dataset : x.S = a) \tag{2.17}$$

$$Entropy = \sum_{c_k \in C} -\frac{\#\{x \in c_k\}}{\#\{x \in Dataset\}} * \log_2 \frac{\#\{x \in c_k\}}{\#\{x \in Dataset\}} \tag{2.18}$$

Where C is the set of existing classes, A is the set of possible values of the attribute S. The $Entropy(Parent)$ is the entropy of the dataset after splitting by the parent attribute. If we are trying to find the root node of the tree, the $Entropy(Parent)$ in the equation 2.16 is the entropy of the original dataset.

The disadvantages of ID3 are: (i) it does not deal with numerical attributes neither missing values; (ii) information gain prefers attributes with more possible values, which can bias the algorithm; and (iii) easily become overfitted.

C4.5 was proposed to overcome the limitations of ID3. C4.5 uses a new metric, called *Gain ratio* [67], that balances the information gain with the number of possible values.

The major advantage of decision trees is their interpretability. It is very easy to understand why a decision tree is classifying an instance as belonging to a specific class.

2.4.4 Support Vector Machines

Support Vector Machines (SVMs) [69] are based on the intuition that the optimal classifier is the one that maximizes the distance between the separating hyperplane and the instances on either side of it. SVMs determine the maximum-margin hyperplane that linearly separates instances from different classes [2].

To understand SVMs it is important to clarify some concepts. Let *w* be an orthogonal vector to the decision boundary, *x* be an arbitrary point with the label *y*, and if *x* is a point in the margin above the decision boundary we will use x_+, and if *x* is a point in the margin below the decision boundary we will use x_-. With all this formalism explained we can now define points in the decision boundary :

$$w \cdot x = c \Leftrightarrow w \cdot x + b = 0 \tag{2.19}$$

By convention, the points in the margin can be define as follows:

$$y(w \cdot x + b) = 1 \tag{2.20}$$

The width of the double margin, can be define as follows:

$$(x_+ - x_-) \cdot \frac{w}{||w||} \tag{2.21}$$

With the width of the double margin defined as in equation 2.21, we still cannot transform this equation into a minimization equation, because we do not know the vectors x_+, x_- and w. However, if we use the information in the equation 2.20, we can rewrite the equation 2.21:

$$\frac{(x_+ \cdot w - x_- \cdot w)}{||w||} \Leftrightarrow \frac{1 - b + b + 1}{||w||} \Leftrightarrow \frac{2}{||w||} \tag{2.22}$$

we can now define a proper optimization problem:

$$\begin{aligned} \underset{w,b}{\text{minimize}} \quad & \frac{1}{2}||w||^2 \\ \text{subject to} \quad & y_i(w \cdot x_i + b) \geq 1, \forall i \end{aligned}$$

This optimization problem can be written using Lagrange multipliers, for each constraint:

$$minL = \frac{1}{2}||w||^2 - \sum_{i=1}^{N} \alpha_i (y_i(w \cdot x_i + b) - 1) \tag{2.23}$$

Deriving the equation 2.23 with respect to w and replace the result, in the next equation, allow us to achieve a friendly classifier equation:

$$f(x) = signal(w \cdot x + b) \Leftrightarrow signal(\sum_{i=1}^{N} \alpha_i y_i (x_i \cdot x) + b) \tag{2.24}$$

The solution must also satisfy this relation :$\sum_{i=1}^{N} \alpha_i y_i = 0$, which is the result of deriving the equation 2.23 with respect to b . The only thing that we still do not know are the values for α, and for that we set to zero the result of deriving the equation 2.23 with respect to w and b. Most of the $\alpha' s$ will turn out to have the value zero, the non-zero $\alpha' s$ will correspond to the support vectors.

The advantages of SVMs are: (i) low tendency to overfitting; (ii) Deal well with high dimensional spaces and (iii) have very great accuracy, in general.

The potentials problems of SVMs are: (i) when the dataset is not linear separable and we need to find a proper kernel function that maps our dataset to a higher dimensional space that are linear separable; (ii) speed and size, both in training and testing dataset [70]; and (iii) it is very difficult to interpret the model.

2.4.5 Neural Network Approach

Neural network approach is inspired in how actually human brain works. Human brain is composed by electrically excitable cells that process and transmit information, these cells are called neurons. In neural approaches one of the simplest algorithm is the single-layered perceptron algorithm [71].

In a single-layered percepton, the perceptron receives as input all the N features $x_1, x_2, ..., x_n$, and then computes the sum of weighted inputs:$\sum_{i=1}^{N} x_i w_i$. If the sum is above a threshold the output is 1; else it is 0. Since the goal is to minimize the error, one possible strategy to train a perceptron is the gradient descent. Considering the mean squared error as:

$$\epsilon = \frac{1}{N} \sum_{i=1}^{N} ||a_i - f(x_i, w, T)||^2 \tag{2.25}$$

Where a_i is the expected class of the i instance, x_i is the i instance, w is the weights and T is the threshold value to activate the neuron. Now we use the gradient descent to actualize the weights (w) and the threshold (T):

$$w = w - \alpha \frac{\partial \epsilon}{\partial w} \tag{2.26}$$

$$T = T - \alpha \frac{\partial \epsilon}{\partial T} \tag{2.27}$$

Where α is the learning rate and measures how greedy we want that the learning process to become.

Perceptron can only classify linearly separable sets of instances. To solve this problem a multi-layer perceptron [72] was proposed.

Multi-layer perceptron consists of three different layers: (i) input layer, which receives the input information; (ii) hidden layer(s), where each hidden layer is composed by multiple percetrons; (iii) output layer, which is a perceptron that receives as input the result of hidden layer(s), and outputs the classification.

One potential problem of multi-layer perceptron is determining the proper size of the hidden layer, if we underestimate the number of neurons we can have poor estimations, but on the other hand, if we have excessive neurons, we can reduce the generalization capability of the network.

To train a multi-layer perceptron the most well known learning algorithm is the backpropagation algorithm [72]:

1. Initialize weights w_{ij} (connecting output of neuron i to neuron j) with random small values.
2. Apply a instance to input layer.
3. Propagate the signal through the network. $z_i = \sigma(w_{ji} z_j)$
4. Calculate the error ϵ.

5. Back propagate δ_i back through the network. $\delta_i = \frac{\partial \epsilon}{\partial net_i}$

6. Actualize the weights. $w_{ij} = w_{ij} + \alpha \delta_i z_i$.

The advantages of neural networks are: (i) very good accuracy; and (ii) works very well with high dimensional data.

The disadvantages are: (i) black box model, very difficult to interpret the model; and (ii) the learning process can takes a lot of time;

2.4.6 Ensemble Methods

In 1995, David Wolpert stated the very famous *no free lunch* theorem: " if an algorithm performs well on a certain class of problems then it necessarily pays for that with degraded performance on the set of all remaining problems". Instead of learning a single very complex model that will only perform well on a certain class of problems, ensemble methods combine the output of several simple classifiers to produce the classification decision.

Random forests [73] are an ensemble method which each classifier is a decision tree (Section 2.4.3) and each tree is grown as follows:

1. If the number of cases in the training set is N, sample N cases at random - but with replacement, from the original data.

2. If there are M input variables, a number $m << M$ is specified such that at each node, m variables are selected at random out of the M and the best split on these m is used to split the node. The value of m is held constant during the forest growing.

3. Each tree is grown to the largest extent possible. There is no pruning.

Each tree is going to vote on a classification and the forest chooses the classification having the most votes.

The advantages of random forest are: (i) very high accuracy; (ii) runs efficiently on large data bases; and (iii) very robust to errors and outliers.

2.5 Discussion

This subsection compares the previously described clustering and classification algorithms, considering the goals of this work and the complexity of clinical data.

To compare the clustering algorithms, we defined seven characteristics that represent, in some way, the goals of this work and the complexity of clinical data:

1. *Speed/Scalability* - Measures the impact of the high dimensionality of the clinical data.
2. *User interpretability* - Measures the interpretability of each algorithm.
3. *User input* - Measures if the algorithm requires input parameters.
4. *Robustness to outliers* - Measures if the algorithm has any mechanism to deal with outliers. Clinical data typically have outliers.
5. *Implementation complexity* - Measures how easy it is to implement the algorithm.
6. *Application to time series* - Measures the existence of related work applying the algorithm to time series.
7. *Model produced* - The model produced is global if each patient in a cluster is defined using all the attributes, and local if each patient in a 'cluster' is defined using a subset of attributes.

Table 2.2 shows a comparison between the clustering algorithms previously described according to these seven characteristics.

The first question that needs to be answered is what family of techniques is more suitable to incorporate in the methodology: clustering or biclustering. Biclustering is only able to produce local models, which means that if we apply this technique we are only able to get groups of patients that show similar evolutions, under a specific subset of features, and under a specific time window. This will result in an enormous amount of clusters, and it will not directly allow us to understand if two patients are or not similar. If we want to find group of similar patients that are in general similar, clustering algorithms seem to be more suitable.

Characteristic	**Biclustering** CCC-Biclust	**Partitioning** K-means	**Hierarchical** Agglomerative	**Model** EM	**Grid** DBSCAN	**Density** STING
Speed/scalability	***	***	*	*	***	**
User interpretability	**	**	***	*	*	*
User input	*	*	***	*	*	*
Robustness to outliers	*	*	*	*	***	***
Implementation complexity	**	***	***	**	*	*
Application to time series	***	***	***	***	*	*
Model Produced	Local	Global	Global	Global	Global	Global

Table 2.2: Comparing clustering algorithms and biclustering. Algorithms considered biclustering: CCC-Biclustering; Partitioning methods: K means; Hierarchical methods: Agglomerative; Model-based methods: EM; Density-based methods: DBSCAN; Grid-based methods: STING. (*** stars represent good and * star represents poor). This table was constructed based on the results present on [2].

DBSCAN and STING are quite complex methods to be implemented and have no significant gain compared to the other methods considering our domain. The only characteristic that these methods are much better than the others is on the robustness to the outliers.

K-means (partitioning method) has the advantage of being very fast, easy to implement and it can be easily applied to time series. However, K-means accuracy is directly dependent on the definition of the centroids and how the centroids are updated. Defining and updating the centroids are challenging issues with no trivial solution available when we are dealing with unequal length time series. Moreover, this method requires input parameters: the number of clusters, and we do not know *a priori* the right number of clusters. Another negative aspect which is not represented in Table 2.2 is the fact that K-means is a non deterministic algorithm, which means that the clusters generated depend on the initialization of the centroids. This need of input and the corresponding non deterministic behavior represent two major drawbacks.

Agglomerative method (hierarchical method) does not have the disadvantages that K-means has. This method does not require an initial number of clusters and does not depend on the definition of the clusters centers. Furthermore, agglomerative method is very well understood. On the other hand, due to the quadratic complexity of the algorithm, agglomerative method is essentially not capable to deal effectively with large time-series, which is a considerable drawback.

EM (model-based method) requires the definition of the number of clusters and has slower processing time than K-means. One potential advantage of model based methods, which is not represented in Table 2.2, is the fact that it is easy to incorporate domain knowledge.

To sum up, all methods have strengths and weaknesses, but since we want that the proposed predictive models to be simple and interpretable, partitioning and hierarchical methods seem to be the a good choice if we need to find groups of similar patients.

To tackle all of the problems of the clinical data, choosing the right clustering algorithm is not enough. We need to carefully choose our similarity measure.

We will compare the different similarity measures considering the following characteristics:

1. *Speed/Scalability* - Measures the impact of the high dimensionality of the EHRs.

2. *Robustness to outliers* - Measures if the similarity measure has any mechanism to deal with outliers.

3. *Applicability to unequal length/sampling time series* - Measures if the similarity measure can deal with unequal length time series and time series with different sampling rates.

4. *Accuracy* - Measures the quality of the similarity measure.

Table 2.3 shows the comparison between the three most used similarity measures. If we do not need to deal with unequal length time series, euclidean distance can be used. if we have to deal with unequal length time series we need to chose between DTW and LCSS. The main difference between DTW and LCSS is that LCSS is typically more robust to outliers than DTW.

Characteristic	Euclidean	DTW	LCSS
Speed/scalability	***	**	**
Robustness to outliers	*	**	***
Applicability to unequal length/sampling time series	-	***	***
Accuracy	**	***	***

Table 2.3: Comparing Euclidean distance, Dynamic Time Warping (DTW) and Longest Common Subsequence (LCSS) (*** stars represent good and * star represents poor). The accuracy evaluation was made by the results present in [3].

Concerning the classification task, we will compare the classification algorithms considering five characteristics that reflect our constraints:

1. *Accuracy* - Measures the proximity of the predicted results to the reality.
2. *Speed of learning* - Measures the impact of the high dimensionality of the clinical data.
3. *Speed of classification* - Measures the time required to classify new instances.
4. *Robustness to outliers* - Measures if the algorithm has any mechanism to deal with outliers.
5. *User interpretability* - Measures the interpretability of each algorithm. One of our requirements is that physicians need to understand the predictions.

Characteristic	Decision Trees	Neural Networks	Naive Bayes	kNN	SVM
Accuracy	**	***	**	**	***
Speed of learning	***	*	***	***	**
Speed of classification	***	***	***	*	***
Robustness to outliers	**	**	***	*	**
User interpretability	***	*	***	**	*

Table 2.4: Comparing different classification algorithms applied to static data (*** stars represent good and * star represents poor), inspired in the table present on [2].

Table 2.4 shows the comparison between the classification algorithms, when the algorithms are applied to static data. One interesting thing of being analyzed is the fact that algorithms with higher accuracy are less understood by the users. SVMs and neural networks are those with higher accuracy, but decision trees, naive bayes and kNN have the advantage of being transparent models, allowing the users to fully understand the classification decision.

The reality expressed in Table 2.4 concerning the accuracy is not correct if we are dealing with time series. The first problem that arises is that we have to deal with unequal length time series and none of these methods, except kNN with a proper distance function, are able to deal with unequal length time

series. One possible solution is to transform unequal length time series into equal length time series, but even with this modification, kNN with DTW is exceptionally difficult to beat [74]. Since we want that the decision can be fully interpretable by the physicians, we think that kNN can be the most suitable method to incorporate in the methodology due to its simplicity and interpretability . For instance, using kNN we can say to the physician that a patient p needs to receive a treatment because he had a similar health condition evolution to a patient z (past patient) that received the treatment.

2.6 Dataset

This study used data from the Medical Information Mart for Intensive Care (MIMIC III) [75]. MIMIC is a large, freely-available database comprising deidentified health-related data associated with over 40,000 patients who stayed in critical care units of the Beth Israel Deaconess Medical Center between 2001 and 2012. The database includes information such as demographics, vital sign measurements made at the bedside ($\sim$ 1 data point per hour), laboratory test results, procedures, medications, caregiver notes, imaging reports, and mortality (both in and out of hospital).

MIIMIC III is a relational database consisting of 26 tables (see Table 2.5). Tables are linked by identifiers which usually have the suffix *ID*. For instance, SUBJECT_ID refers to a unique patient, and ICUSTAY_ID refers to a unique admission to an intensive care unit. Further information of MIMIC III can be find in [4].

Table 2.5: Overview of the 26 tables of MIMIC III [4].

Table Name	Description
ADMISSIONS	Every unique hospitalization for each patient in the database (defines HADM_ID).
CALLOUT	Information regarding when a patient was cleared for ICU discharge and when the patient was actually discharged.
CAREGIVERS	Every caregiver who has recorded data in the database (defines CGID).
CHARTEVENTS	All charted observations for patients.
CPTEVENTS	Procedures recorded as Current Procedural Terminology (CPT) codes.
D_CPT	High level dictionary of Current Procedural Terminology (CPT) codes.
D_ICD_DIAGNOSES	Dictionary of International Statistical Classification of Diseases and Related Health Problems (ICD-9) codes relating to diagnoses.
D_ICD_PROCEDURES	Dictionary of International Statistical Classification of Diseases and Related Health Problems (ICD-9) codes relating to procedures.
D_ITEMS	Dictionary of local codes ('ITEMIDs') appearing in the MIMIC database, except those that relate to laboratory tests.
D_LABITEMS	Dictionary of local codes ('ITEMIDs') appearing in the MIMIC database that relate to laboratory tests.
DATETIMEEVENTS	All recorded observations which are dates, for example time of dialysis or insertion of lines.
DIAGNOSES_ICD	Hospital assigned diagnoses, coded using the International Statistical Classification of Diseases and Related Health Problems (ICD) system.
DRGCODES Diagnosis	Related Groups (DRG), which are used by the hospital for billing purposes.
ICUSTAYS	Every unique ICU stay in the database (defines ICUSTAY_ID). INPUTEVENTS_CV Intake for patients monitored using the Philips CareVue system while in the ICU, e.g., intravenous medications, enteral feeding, etc.
INPUTEVENTS_MV	Intake for patients monitored using the iMDSoft MetaVision system while in the ICU, e.g., intravenous medications, enteral feeding, etc.
OUTPUTEVENTS	Output information for patients while in the ICU. LABEVENTS Laboratory measurements for patients both within the hospital and in outpatient clinics. MICROBIOLOGYEVENTS Microbiology culture results and antibiotic sensitivities from the hospital database.
NOTEEVENTS	Deidentified notes, including nursing and physician notes, ECG reports, radiology reports, and discharge summaries. PATIENTS Every unique patient in the database (defines SUBJECT_ID). PRESCRIPTIONS Medications ordered for a given patient.
PROCEDUREEVENTS_MV	Patient procedures for the subset of patients who were monitored in the ICU using the iMDSoft MetaVision system. PROCEDURES_ICD Patient procedures, coded using the International Statistical Classification of Diseases and Related Health Problems (ICD) system.
SERVICES	The clinical service under which a patient is registered.
TRANSFERS	Patient movement from bed to bed within the hospital, including ICU admission and discharge.

3

Methodology for the Construction of Medical Predictive Models

Contents

This chapter describes the methodology followed in the construction of the two predictive models proposed in this work for the prediction of vasopressors administration and mechanical ventilation. Given a database comprising health-related data of patients, we propose a methodology composed by well defined steps that suggest actions to use the data stored in this type of database in order to create a predictive model capable of assisting physicians when they are making decisions. The methodology encompasses four steps: data pre-processing (Section 3.1), selection of the patient representation (Section 3.2), feature ranking (Section 3.3) and construction of the predictive model (Section 3.4).

3.1 Data Pre-Processing

The data pre-processing that must be applied to a dataset always depends on the characteristics of the data and the problem that we want to solve. However, there are four tasks that are usually required: selection of patient records, identification of relevant features, selection of feature measurements, and data normalization.

The first task to be performed is the *selection of patient records*. First, we need to define the patient profile that makes sense to be considered according to the problem domain. For instance, if we want to predict the application of a treatment that can only be applied to adults, our patient profile should be adult patients. Once we have clearly defined the patient profile, we must select the records of the patients that are in accordance to that profile.

Then, it is mandatory to proceed with the *identification of relevant features*. The goal is to select among the features that we have for all patients (e.g., heart rate, urine output) those that are relevant for the problem in consideration. For example, if the target is to predict the mortality of patients inside intensive care units, heart rate is probably an important feature for a predictive model. If we want to predict the need of hemodialysis, the heart rate can be probably ignored. The incorporation of domain knowledge is very important in this task. In case of doubt or ignorance we should always consider a larger set of features, because the early exclusion of relevant features will have a negative impact in a future predictive model.

Most of the data stored in a health-related database are inserted by humans. Humans are error prone, so several measurements stored may be wrong. Therefore a *selection of feature measurements* must be performed in order to eliminate invalid measurements. For all features considered, all measurements whose value is not contained in a pre-defined range must be ignored. For instance, if we have a negative body temperature measurement for a patient, we must ignore this measurement, because such value is not possible. In order to define the possible value ranges for each feature, domain knowledge must be incorporated. We can also apply other methods, such as interquartile range methods, where we remove the measurements that are 1.5 interquartile ranges (quartile 3 - quartile 1) below the first

quartile or above the third quartile. Given the following measurement values for a feature A, $\{$2, 5, 4, 3, 12, 5, 3, 4, 3, 2, 3$\}$, the first quartile is 3 and the third quartile is 4.5, so all the measurement values that are not inside $[0.75, 6.75]$ are ignored. The measurement with value 12 is going to be ignored, because it is outside the interval. The problem of this method is that we are ignoring some measurement values that are possible in real world and our future predictive model should take into account such cases.

Finally, it is important to perform *data normalization*. Each feature can have different ranges of measurement values. This diversity may cause problems in a future predictive model. For example, glucose measurements can vary from 0 to 10000 mg/dL and body temperature measurements can vary from 0 to $50°$C. There are different techniques to normalize data, one of the simplest is the *min-max normalization*, where, for each measurement value of a feature, we apply the following formula:

$$y = \frac{x - min(X)}{max(X) - min(X)} \tag{3.1}$$

where X is the set of all measurement values of a feature f and x is a measurement value from that set. With *min-max normalization* all measurements will be between 0 and 1.

3.2 Selection of the Patient Representation

For each patient most of the features correspond to a collection of measurement values collected during an interval of time (temporal data). Temporal data is complex to handle because the amount of data is different from patient to patient and the data are collected at a wide variety of sampling frequencies. The goal of this step is to solve the problems associated with the data in a patient representation.

We propose a *closed window representation* for patients. The idea of this representation is to split the time interval of each patient in N equal sized windows. Each window is represented using one or more aggregation functions that are applied to all measurements in that window, for each feature. The average is an example of an aggregation function, but there are plenty of other options. Table 3.1 shows a list of possible aggregation functions.

Table 3.1: Examples of aggregation functions applied to a window with the following the set of measurement values $x = [x_1, x_2, x_3, ..., x_n]$

Aggregation Function	Description	Formula
1	average	$\frac{1}{n}\sum_1^n x_n$
2	median	$\tilde{x}$
3	standard deviation	σ_x
4	maximum value	$max_n x_n$
5	location of maximum normalized	$\frac{1}{n} argmax_n x_n$
6	minimum value	$min_n x_n$
7	location of minimum normalized	$\frac{1}{n} argmin_n x_n$
8	average absolute change	$\frac{1}{n}\sum_2^n \lvert x_n - x_{n-1}\rvert$
9	average change	$\frac{1}{n}\sum_2^n x_n - x_{n-1}$
10	Ordered weighted averaging	$\sum_1^n x_n * w_i$

In order to show the applicability of this representation, Table 3.2 exemplifies the representation of a patient *p* with one feature (heart rate) using four windows of one hour each and considering only the average as the aggregation function.

Table 3.2: Representation of a patient with one feature (heart rate), using four windows of one hour each and the average as the aggregation function.

Windows	Time	Value	Window Average
window 1	00:00H 00:30H	92 94	93
window 2	01:00H 01:15H 01:30H	94 95 96	95
window 3	-	-	95
window 4	03:00H	100	100

Using the proposed representation in this small example, we go from a six dimensional space (six measurement values) represented in the column Value to a four dimensional space (four windows) represented in the column Window Average. In real world, the number of measurement values per feature is much bigger so this patient representation enables to achieve a significant reduction of the dimensional space and solves the problem of having features with different collection frequencies.

With the *closed window representation* several windows may not contain any measurement value for some features. In these cases the aggregation functions assigned to the previous window are assigned to the window without measurement values. By doing this, we are assuming that the evolution remained constant in this period. In Table 3.2, this technique was applied to window 3. Since this window did not have any measurements, the result of the aggregation function of window 2 was assigned to window 3. We can also apply a linear regression to predict the results of the aggregation functions of windows without measurement. With this technique we cannot consider the future windows when we are applying a linear regression, because if we use the future information to fill the windows without measurements, the predictive model based on this representation will be biased.

One challenge of the *close window representation* is to find the right size of the window, because if we choose a large window, the aggregation functions chosen may not be enough to represent all measurement values contained in each window. If we choose a small window, this may result in several windows without measurement values. Choosing the window size should be based on the sampling frequency of the different features and should be a trade-off between the most frequent features and the least frequent features. Different sizes of window should be assessed in order to find the best window size.

The *closed window representation* is just one possibility, many other representations can be chosen. For instance, we can see the evolution of each patient as transitions between states. In this case, each patient can be represented by the set of states through which he has passed or the probability of being in each state [6].

3.3 Feature Ranking

The goal of this step is to assess the importance of each feature for the prediction based on the characteristics of the data.

To assess the importance of each feature we want to answer the following question: *What are the features that better distinguish the different types of patients?* An example of two types of patients is the patients who received a treatment and the patients who did not receive a treatment. In order to answer this question two aspects need to be clarified. First, we need to define how to divide the population of patients by each feature. Second, we need to define how to evaluate the quality of the division chosen for each feature.

In clinical data we have essentially two types of features: static (e.g., age) and temporal (e.g., heart rate). Since the nature of these two types of features is different we apply different division techniques. For static features, we apply an equal height discretization, so we will find groups of patients that have the same number of patients. For temporal features, we have several measurements in time, so we want to find groups of patients that had a similar evolution. A simple way of finding groups of patients in this type of data is through clustering techniques. With clustering techniques we are able to find groups of patients that show a similar evolution for each feature. Figure 3.1 shows a visually separation of glucose in N groups of patients (clusters). Clusters are going to have different amounts of patients of the different types. Ideally we want that each cluster only has patients of a specific type.

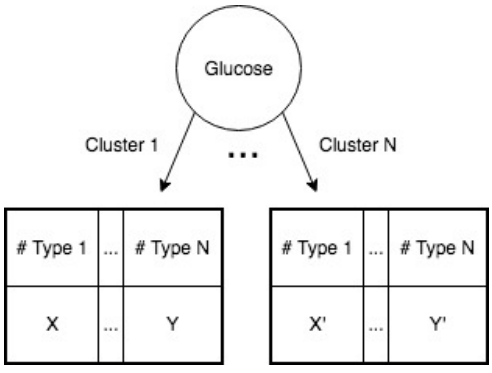


Figure 3.1: Division of glucose in N clusters. The tables represented reflect the counters for each patient type inside each cluster.

In Section 2.1 we described different clustering algorithms. In a case of low computational power, partitioning methods (e.g., K-means) are the most suitable algorithms to find the groups of patients for each temporal feature, because these methods are usually faster and simpler than the other methods.

To evaluate the quality of the division chosen for each feature we calculate the *information gain ratio* [67]. The *information gain ratio* balances the *information gain* with the number of groups considered. From a mathematical point of view the *information gain ratio* of a feature F is given by the following

equation:

$$InformationGainRatio(F) = \frac{InformationGain(F)}{\sum_{k=1}^{K} Entropy(\{x \in Dataset : x.F = cluster_k\})} \tag{3.2}$$

The meaning and equations of *Information Gain* and *Entropy* are described in Section 2.4.3.

For each feature we need to find the number of groups (clusters) that maximizes the *information gain ratio*. To do this, we successively consider larger groups. At the end, we choose the highest value obtained as the importance of that feature.

3.4 Construction of the Predictive Model

The goal of this step is to build a predictive model capable of assisting physicians when they making clinical decisions. In Section 2.4 we described some machine learning algorithms that can be applied in this context. The algorithm chosen should allow physicians to interpret the predictions. A fairly simple predictive algorithm that can be easily interpreted by physicians is kNN. This method has the advantage of exploring the natural intuition of similarity between patients. For instance, it would be interesting to inform a physician that a patient p needs to receive vasopressors because she has a similar evolution to a patient z that received the treatment. Moreover, we can increase the interpretability of the kNN results by applying the algorithm on each feature separately (local prediction) and considering k=1. Then, the final prediction is based on the local predictions.

There are different ways of combining the local predictions to produce a final prediction. One of the simplest techniques is to choose the most frequent prediction among the locals predictions as the final prediction. For instance, imagine that we are considering just three features: heart rate, glucose and respiration rate, and we want to predict the type of a patient *p*. Using a simple kNN with k equals to one, we discovered on the clinical repository that patient p is similar to a patient x on the heart rate and patient x is *Type 1*. On glucose, patient p is similar to a patient y and patient y is *Type 2*. Finally, on respiration rate patient p is similar to a patient z and patient z is *Type 1*. Since *Type 1* was the most frequent type among the three different features, the final prediction of patient p will be *Type 1*.

Figure 3.2 shows the context provided to physicians to interpret the prediction of patient p. In this example, associated with the prediction of patient p (Type 1) we give to the physician the evolution of heart rate of patient x and the evolution of respiration rate of patient z and say that both patients are Type 1. Moreover, we can alert the physician that according to the evolution of glucose levels, patient p should be Type 2. With this additional information based on the past patients, we claim that the physicians will have the necessary information to accept or reject the prediction.

Other possibility for combining the local predictions is to use the importance of the features calculated on the previous step of the methodology (feature ranking). Instead of being just the most frequent

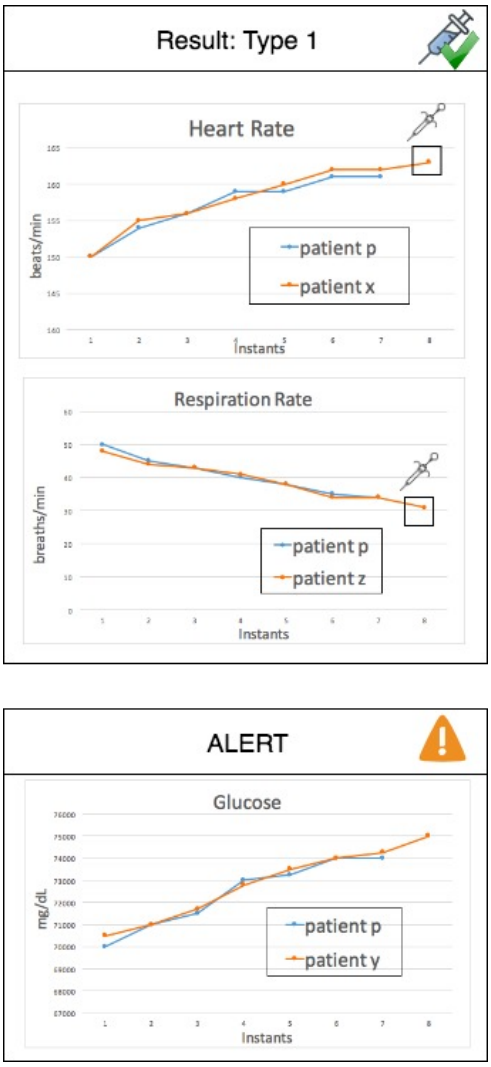


Figure 3.2: Predicting the type of a patient using kNN applied to each feature separately.

prediction among the local predictions we can weight the local predictions by their importance (normalized). For instance, in the prediction of patient p if the importance of glucose is 0.8 and the importance of heart rate and respiration rate is just 0.1, the final prediction for patient p will be Type 2, because glucose is very important for the prediction.

One crucial step in the construction of any predictive model is the feature selection. We can do a feature selection based on importance of the features calculated on the previous step (feature ranking). We start by considering a set of features composed by the most important feature and we use that set of features on the model. Then, we successively evaluate and add to the set of features the next feature more important. In the end, we can return the model with the set of feature that gives better results or that gives results slightly lower than the best result but uses less features. The smaller the set of features used, more interpretable the predictions will be.

4

Prediction of Vasopressors Administration

Contents

In this chapter we apply the methodology previously described to construct a predictive model capable of assisting physicians when they are prescribing vasopressors administration. We start by producing a predictive model capable of making predictions for the next hour. We apply the four steps of the proposed methodology: data pre-processing (Section 4.1), selection of patient representation (Section 4.2), feature ranking (Section 4.3) and construction of predictive model (Section 4.4). We also construct a predictive model capable of making predictions two hours in advance in order to compare the results with the state of the art.

4.1 Data Pre-Processing

In this step we performed four tasks: (i) selection of patient records, (ii) identification of relevant features, (iii) selection of feature measurements, and (iv) data normalization.

Selection of patient records: To select the patient records we need to define the patient profile that we want to consider. Since vasopressors are a treatment administrated usually in adults we only considered adult patients. Moreover, we wanted to guarantee that the patients stayed in the intensive care units for a reasonable amount of time to avoid non sick patients and to have a minimal amount of clinical data about the patients. So, we only considered patients that stayed at least 24 hours in the intensive care units (ICU). For patients who received vasopressors, we considered only those who had at least 8 hours of clinical data before they started receiving vasopressors. We only selected the patients that satisfied this profile.

Identification of relevant features: Since we did not have the support of a physician to help us selecting the most appropriate features, we selected 24 initial features (Table 4.1) based on the set of features used in similar works [5, 6, 8, 9]. We did not add more features, because that would make us lose more patients. So, as a trade off between more patients versus more features we selected this final subset of features. We leave for future work the analysis of other potential subset of features that can include, for instance, the glasgow coma scale or the weight of each patient. The extraction of the medical features and the information about the application of vasopressors was applied directly to MIMIC-III through SQL scripts. The SQL scripts were adapted from the scripts provided by the team responsible for the maintenance of the MIMIC repository [10].

The first two tasks resulted in the following patient characteristics: **Criteria 1:** patients with age $>$ 15, to exclude pediatric patients[1]; **Criteria 2:** in the case of multiple ICU's admissions, we only considered the first admission to avoid later developed complications; **Criteria 3:** patients containing at least one measurement of all features presented in Table 4.1; **Criteria 4:** patients who stayed in the ICU at least 24 hours after the instant where they had at least one measurement of all features (Table 4.1) ; and **Criteria**

[1] In several works that used MIMIC, patients with age higher than 15 years are seen as adults

Table 4.1: List of the 24 initial features considered

Feature	Unit	Category	Acceptable interval
Heart Rate	$beats/min$	Vital sign	[0, 300]
Temperature	$Celsius$	Vital sign	[0, 50]
SpO_2	$\%$	Vital sign	[0, 101]
Respiratory Rate	$breaths/min$	Vital sign	[0, 70]
Non-invasive systolic blood pressure	$mmHg$	Vital sign	[0, 400]
Non-invasive diastolic blood pressure	$mmHg$	Vital sign	[0, 300]
Non-invasive mean blood pressure	$mmHg$	Vital sign	[0, 100]
Hematocrit	$\%$	lab test	[0, 1000]
White Blood Cells	$10^3/\mu L$	lab test	[0, l000]
Platelets	$K/\mu L$	lab test	[0, l0000]
Hemoglobin	g/dL	lab test	[0, 50]
Potassium	mEq/L	lab test	[0, 30]
Sodium	mEq/L	lab test	[0, 200]
Chloride	mEq/L	lab test	[0, 10000]
Bicarbonate	mEq/L	lab test	[0, 10000]
Anion Gap	mEq/L	lab test	[0, 10000]
BUN	mg/dL	lab test	[0, 300]
Creatinine	mg/dL	lab test	[0, 150]
Glucose	mg/dL	lab test	[0, 10000]
INR	$ratio$	lab test	[0, 50]
PT	sec	lab test	[0, 150]
PTT	sec	lab test	[0, 150]
Age	$year$	static	-
Gender	$binary$	static	-

5: patients who had at least 8 hours of clinical data before they received vasopressors. In this study we considered the application of seven types of vasopressors: norepinephrine, epinephrine, phenylephrine, vasopressin, dopamine, dobutamine, and milrinone. A flowchart of the inclusion procedure is depicted in Figure 4.1.

Selection of feature measurements: For all features considered, we excluded all measurements that were not inside the acceptable interval described in Table 4.1. This acceptable interval per feature was defined by the team responsible for the maintenance of MIMIC.

Data normalization: The measurement values of all features were normalized using the *min-max* normalization technique, setting the range of all features to [0, 1].

4.2 Selection of the Patient Representation

In this work, we used the *closed window representation* described in Section 3.2. The time interval of measurements for each patient was divided in N temporal windows of one hour. Each temporal window was represented by the average of all measurements contained in that window. The size of the window was based on the fact that the vital signs were measured every hour in the worst case. We let for future work the study of different window sizes.

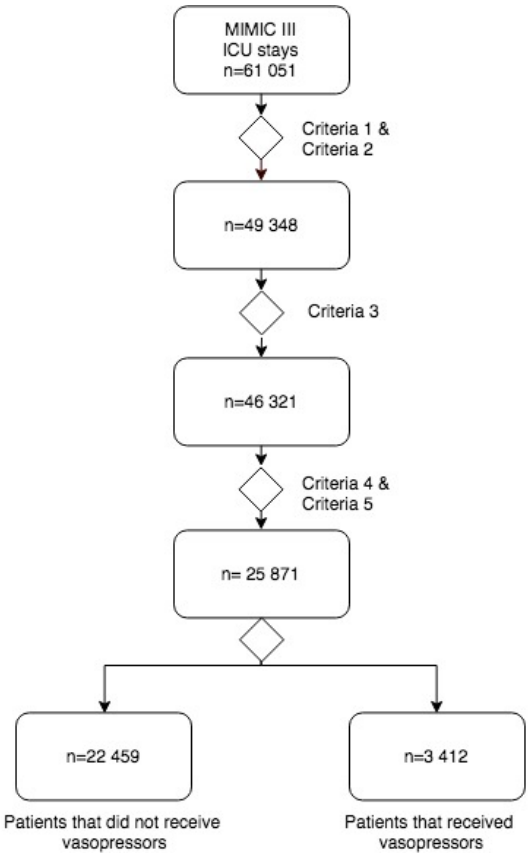


Figure 4.1: Patient selection flowchart

4.3 Feature Ranking

The goal of this step was to assess the real importance of the features on the prediction of vasopressors administration.

For patients of both types (patient who received vasopressors and patients who did not receive vasopressors), we analyzed the importance of the features in two contexts: (i) considering the data before the patients received vasopressors, and (ii) considering the data before and after the patients received vasopressors.

In the first context, we considered 8 windows out of the N windows that represented each patient. For patients who received vasopressors, the windows were the 8 windows before the patients received vasopressors (including the window where the patient received vasopressors). For patients who did not receive vasopressors, we extracted randomly 8 consecutive windows. By doing this, we had the same amount of information for both types of patients.

In the second context, we considered 12 windows out of the N windows that represented each patient. For patients who received vasopressors, the 12 windows were composed by the 8 windows before they received vasopressors (the same windows used in the first context) and the 4 windows after

they received vasopressors. For patients who did not receive vasopressors, we extracted 12 consecutive windows randomly. In both contexts, we expected that the features that justified the administration of the treatment should be those that had a ”different” evolution between both types of patients.

The number of windows considered in both contexts (8 and 12 windows) was based on the fact that the number of windows considered affects the number of patients in our dataset. If we considered more than 8 windows before the patients received vasopressors, the number of patients that respected this constraint would be smaller. So, again as a trade-off between more information versus less patients we used 8 windows before the patients received vasopressors. In the second context (12 windows) we used four windows after the patients received vasopressors, because it seemed to us a reasonable amount of information.

The clustering algorithm chosen to find groups of patients with similar evolution within each feature was *K-means*. We started by identifying for each feature the number of groups of patients (clusters) that maximizes the *information gain ratio* (Equation 3.2). The importance of each feature is the maximum *information gain ratio*. Table 4.2 shows the features with higher values of the *information gain ratio* and the number of patient groups that maximized the value, considering eight hours of data before the patients received vasopressors.

Table 4.2: Information gain ratio of features with higher value considering 8 hours of data.

Feature	***Information Gain Ratio*** **(Importance)**	Groups (K)
SysBP	0,02122376	3
MeanBP	0,01150347	3
DiasBP	0,01090331	2
RespRate	0,00579126	3
TempC	0,005741323	3
PTT	0,004474779	4
WBC	0,004397975	3
HeartRate	0,003444196	3
BUN	0,002873469	2
Potassium	0,002331212	4
SpO2	0,002108554	4
Anion Gap	0,002100305	2
Bicarbonate	0,002043527	4
Creatinine	0,001388004	4
Hemoglobin	0,001269939	3
Platelet	0,001255194	3
Sodium	0,00112645	2

Observing Table 4.2 we concluded that the systolic blood pressure (SysBP), the mean blood pressure (MeanBP) and the diastolic blood pressure (DiasBP) were the features with higher *information gain ratio*. The importance revealed by these features is aligned with the main goal of vasopressors administration, which is to increase the arterial pressure. With a simple procedure based on clustering and the *information gain ratio*, we were capable of finding a feature ranking that seems to be related with the administration of vasopressors.

Table 4.3 shows the features with higher values of *information gain ratio* and the number of groups

Table 4.3: Information gain ratio of features with higher value considering 12 hours of data.

Feature	*Information Gain Ratio* (Importance)	Groups (K)
SysBP	0,02630749	3
DiasBP	0,01355699	2
MeanBP	0,01185054	2
PTT	0,005472286	3
WBC	0,005426762	3
TempC	0,005303975	4
RespRate	0,003342048	3
Bicarbonate	0,003200597	2
BUN	0,002913539	2
HeartRate	0,002826552	3
Anion Gap	0,002785657	2
Potassium	0,002499508	2
SpO2	0,001900655	3
INR	0,001697482	4
Creatinine	0,00160697	3
PT	0,001364693	3
Hematocrit	0,001282766	5

that maximized the result considering 12 hours of data (8 hours before and 4 hours after the patients received vasopressors). The three most important features were the same, although with the usage of 12 hours of clinical data, the diastolic blood pressure had more importance than the mean blood pressure. The significant differences between both tables started to appear after this *top* 3. For instance using 8 hours of data the respiration rate was the fourth most important feature, while using 12 hours of data the respiration rate was the seventh most import feature.

4.4 Construction of the Predictive Model

In the description of this step (Section 3.4) we saw that a model based on kNN applied to each feature individually can be easily interpretable by physicians. To create such model several aspects need to be tested in order to find the best parameterization. We decided to perform four experiments with different goals: **Experiment 1** - evaluation of the impact of applying kNN to all features together versus applying kNN to each feature individually (the most frequent prediction among the local predictions is the final prediction); **Experiment 2** - evaluation of the proposed feature selection procedure based on the importance of features versus an automated feature selection procedure (sequential forward selection); **Experiment 3** - evaluation of the impact of weighting the local predictions by the importance of the features; **Experiment 4** - evaluation of the impact of using on the feature ranking step different amounts of hours. In all experiments we considered always k=1 in the kNN[2].

We started to perform these 4 experiments in order to find the best model to predict the administration of vasopressors within 1 hour. Then, we apply the best model to predict the administration of

[2] Parameterization: distance function: Euclidean, k=1, R package: [76]. This parameterization is the same for all tests that applied kNN.

vasopressors within 2 hour.

This section is organized as follows: We start by describing the experimental setup followed in all experiments (Section 4.4.1). Then we create our baseline based on a neural network (Section 4.4.2). Then we analyze the results of each experiment separately (Section 4.4.3 - 4.4.6). And at the end of this section we specify the final model achieved for one and two hours predictions, and we compare the results with the state of the art (Section 4.4.7).

4.4.1 Experimental setup

Our experimental population consisted on 22 459 patients who did not receive vasopressors and 3 412 patients who received vasopressors. For patients who received vasopressors, we extracted the 7 windows immediately before the window where patients received vasopressors, so that we could do predictions one hour in advance. For patients who did not receive vasopressors, we extracted randomly 7 consecutive windows of the first stay. By doing this, we used the same amount of information for both types of patients.

To construct our dataset, we randomly sampled the patients who did not receive vasopressor (negative class), selecting one negative class for each positive class (patients who received vasopressors) without replacement. This resulted in a dataset with 6 824 patients where 50% received vasopressors and 50% did not receive vasopressors. We also guaranteed that on this dataset all patients were unique, no patient was represented twice.

To assess the performance of a model, we applied a 10-fold cross validation process on the sampled dataset. However, since the number of patients who did not receive vasopressors was seven times higher than the number of patients who received vasopressors, assessing the performance of a model based only on a single sample dataset was not enough. So, to evaluate the performance of the model we repeated the following process ten times: (i) construction of a dataset where the number of patients of both types (patient who received vasopressors and patients who did not receive vasopressors) was the same; and (ii) application of 10 fold cross validation. Repeating this process 10 times for each experiment returned 100 results (10 datasets * 10 fold cross validation). At the end, we calculated the average sensitivity, specificity and AUC considering all results. Figure 4.2 summarizes the experimental setup.

4.4.2 Creation of Baseline

As baseline we applied a neural network[3] to all features together (the initial 24 features). This is a well known method that typically presents high accuracy. However, this method is a ”black box” with

[3]Parameterization: single hidden layer with 5 Neurons, learning algorithm - BFGS algorithm [77], iterations - 200, decay - 5e-4, rang - 0.1, activation fun - logistic, error - least squares, R package - nnet [76].

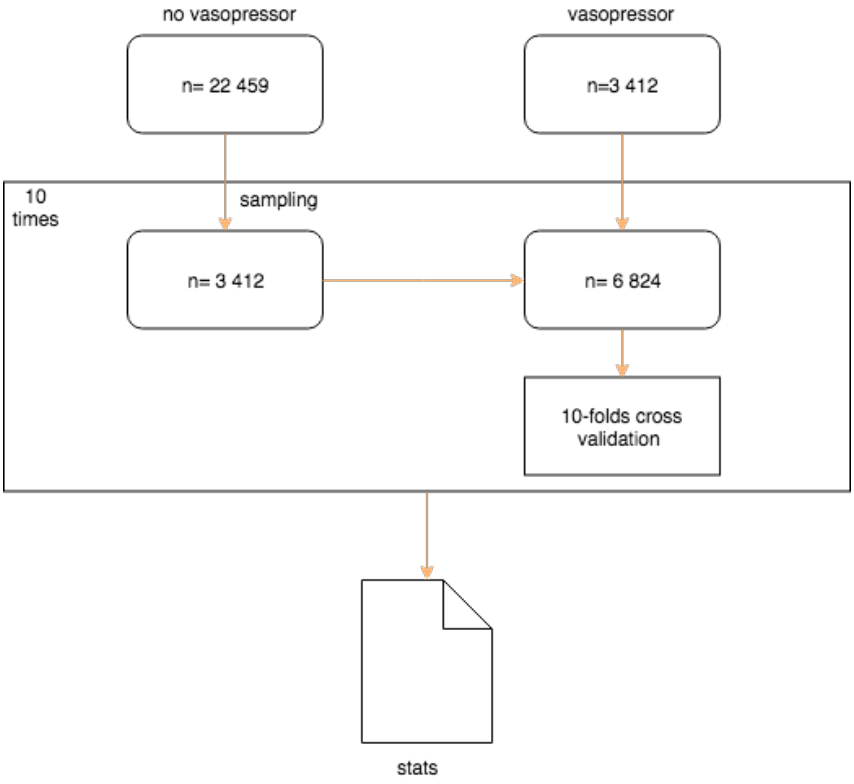


Figure 4.2: Experimental setup flowchart

almost zero interpretability.

Applying a neural network to all features together resulted in an AUC of 0.879. This was a very positive result to start and an evidence that it is possible to achieve good results in the prediction of vasopressors administration.

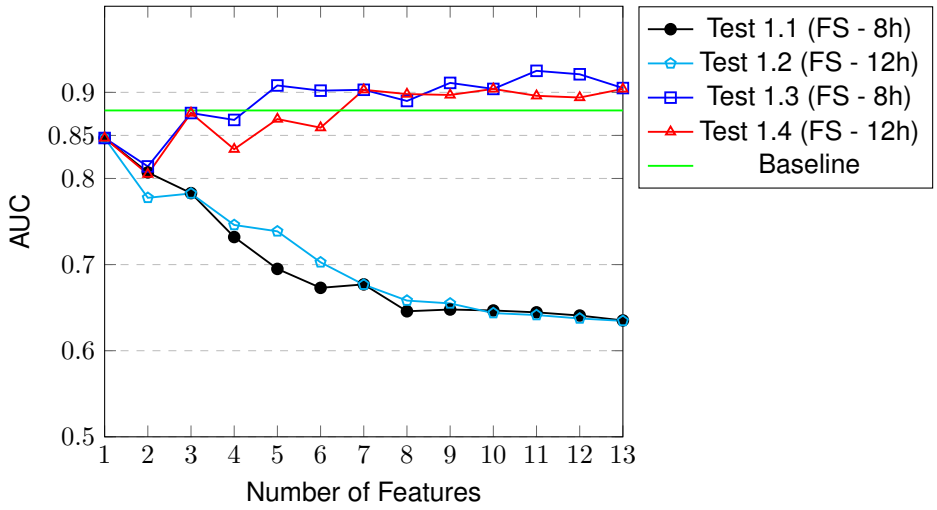
4.4.3 Experiment 1 - Impact of Applying kNN to Each Feature Individually

In this experiment we did 4 tests: **Test 1.1 - kNN applied to all features together** using a feature selection based on the feature ranking calculated using the 8 hours of clinical data before the patients received vasopressors (Table 4.2); **Test 1.2 - kNN applied to all features together** using a feature selection based on the feature ranking calculated using the 8 hours before and the 4 hours after patients received vasopressors (Table 4.3); **Test 1.3 - kNN applied to features individually** using a feature selection based on the feature ranking calculated using the 8 hours of information before the patients received vasopressors (Table 4.2). The final prediction is the most frequent prediction among the local predictions; **Test 1.4 - kNN applied to features individually** using a feature selection based on the feature ranking calculated using the 8 hours of information before and the 4 hours after the patients received vasopressors (Table 4.3). The final prediction is the most frequent prediction among the local

predictions. As we described in Section 3.4 the feature selection based on the feature ranking works as follows: we start with the most import of feature and we add in each step (to the set of features considered) the next feature more important according to the feature ranking.

Figure 4.3 shows the results of this experiment. Observing Figure 4.3 we conclude that: (i) applying kNN to each feature individually (Test 1.3 and 1.4) outperformed kNN applied to all features together (Test 1.1 and 1.2); (ii) the feature selection process for kNN applied to all features together (Test 1.1 and 1.2) only had a positive impact on the choice of the first feature, from that point forward the results were always worst; (iii) the feature selection process based on the feature ranking calculated from using the 8 hours of clinical data before the patients received vasopressors worked better (Test 1.3 versus Test 1.4); and (iv) using only 5 features (systolic blood pressure, mean blood pressure, diastolic blood pressure, respiration rate and body temperature) on a kNN applied to each of these features individually (Test 1.3) outperformed our baseline (neural network) achieving an AUC of 0.908.

Figure 4.3: Experiment 1 - Results



4.4.4 Experiment 2 - Impact of Automated Feature Selection

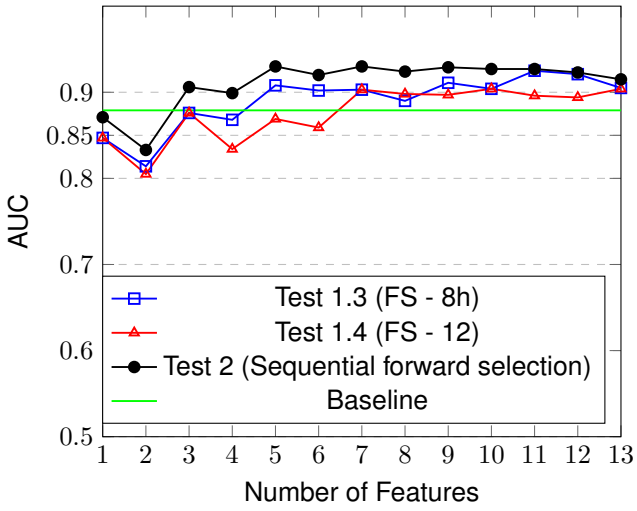
One crucial step in any predictive model is the feature selection step. In this section, we want to evaluate the impact of using an automated feature selection method, the sequential forward selection, where we start with an empty set of features and we add in each step the feature that maximizes the AUC. We did the following test: **Test 2 - apply the sequential forward selection method to a kNN considering each feature individually.** The final prediction is the most frequent prediction among the local predictions. Table 4.4 shows the feature order followed by this method and Figure 4.4 shows the results of this experiment.

This experiment allows us to make some conclusions: (i) the sequential forward selection resulted in

Table 4.4: Best feature order according to feature forward selection for the prediction of vasopressors administration within one hour.

Feature
SpO2
Bicarbonate
SysBP
RespRate
MeanBP
Sodium
HeartRate
Chloride
DiasBP
TempC
PTT
Age
Hematocrit
Creatinine

Figure 4.4: Experiment 2 - Results



a completely different order of features when compared to both feature rankings calculated in the previous step of the methodology. According to the sequential forward selection, features like bicarbonate or SpO2 are very important, which is strange from our point of view, because these features do not seem to be related with the arterial pressure; and (ii) with the sequential forward selection we achieved slightly better results. Nevertheless the difference between using this method (Test 2) and using a feature selection based on the feature ranking (Test 1.3) was not significant. Since the results achieved were not sufficiently different to say if one method is definitely better than the other, we can use other two comparison criteria: time and the set of features used. Despite the sequential forward selection method take more time to run (we need to perform all combinations of features in each step), time is not a problem for us, because we only need to perform this method once. However if we had a larger set of features (e.g. 1000) this method could take more than a week to execute, which could be a problem even if we just need to execute it once. The aspect that can make us choose one method instead of the other

is the set of features used by the model. The top 3 features for the sequential forward selection were: SpO2, Bicarbonate and SysBP, while for the feature selection based on the feature ranking considering the 8 hours of information before the patient received vasopressors, the top 3 were: SysBP, MeanBP and DiasBP. Physicians can only understand a prediction if it is based on features that they already know about their relevance. So, the method whose feature set is more appreciated by the physicians is the best method. For us the sequence of features from using the feature ranking makes more sense, but we need to validate this intuition with physicians. Table 4.5 compares the results of the three feature selection processes.

Table 4.5: Comparison of the three features selection processes

Feature selection	First AUC $>$ Baseline	Best AUC
Feature ranking (8 hours)	0.908$\pm$0,011 (5 features)	0.925$\pm$0.011 (11 features)
Feature ranking (12 hours)	0.903$\pm$0,011 (7 features)	0.904$\pm$0,013 (10 features)
Sequential forward selection	0.906$\pm$0.012 (3 features)	0.930$\pm$0.009 (5 features)

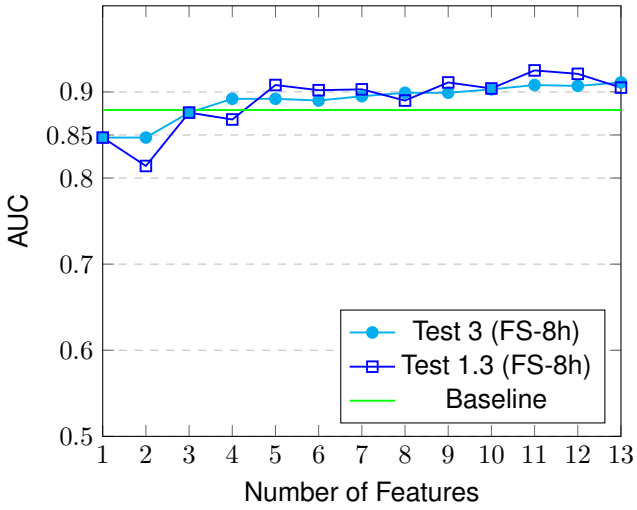
4.4.5 Experiment 3 - Impact of Weighting the Predictions

In the previous step of the methodology we quantified the importance of each feature for the prediction of vasopressors administration. We can use the importance of the features (*information gain ratio*) to weight the local predictions. In other words, instead of applying a kNN to each feature separately and derive a final prediction based on the most frequent prediction among these local predictions, we can weight these local predictions by their importance. Moreover, in the Experiment 2 we can see an up and down behavior of the AUC with the increase in the number of features (Figure 4.4). If the number of features is even, we see a decrease of the AUC, if the number of features is odd we see an increase of the AUC. This behavior results from the fact that we are having a lot of ties when the number of features is even. When we have ties, the final prediction is random, which justifies this up and down behavior, specially when the number of features is small.

To evaluate the impact of weighting the features, we performed one test: **Test 3 - kNN applied to each feature individually weighted by the importance of the features (see Table 4.2)**. We also applied a feature selection based on the feature ranking resulted from using the 8 hours of clinical data before the patients received vasopressors (Table 4.2). We just considered this feature ranking, because it was what returned better results in the previous experiments. Figure 4.5 shows the results of this experiment.

This experiment allows us to make a few conclusions: (i) the up and down behavior gave rise to an almost linear growth of the AUC with the increase of the number of features considered; and (ii) weighting the local predictions only had a positive impact considering a small set of features.

Figure 4.5: Experiment 3 - Results.



4.4.6 Experiment 4 - Impact of Repeating the Feature Ranking Step

In Experiment 2 we saw that a feature selection based on the feature ranking resulted from using the 8 hours of clinical data before the patients received vasopressors returned similar results when compared to the sequential forward feature selection. A natural question to ask is: What happened if instead of considering 8 hours of clinical data (before the patients received vasopressors) we considered less hours?. To answer this question we repeated the feature ranking step considering six and four hours/windows of information before the patients received the vasopressors. Table 4.6 shows the ranking of features resulted from using 6 and 4 hours/windows on the feature ranking step (from the most important feature to the least important feature).

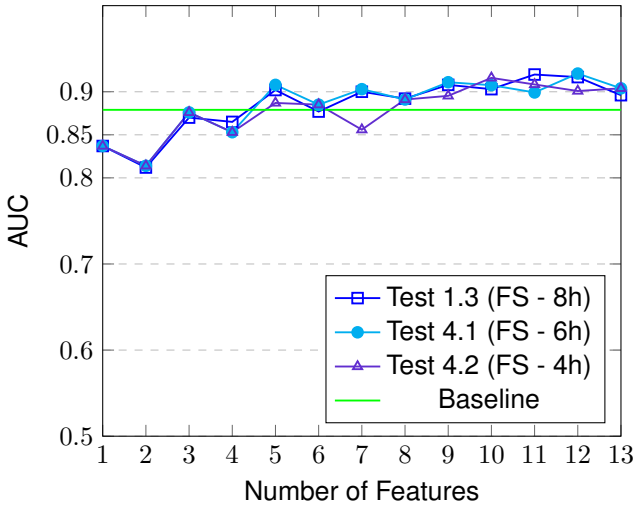
Table 4.6: Best feature ranking according to feature ranking step using six and four windows before the patients received vasopressors.

Features 8 windows	Features 6 windows	Features 4 windows
SysBP	Sysbp	Sysbp
Meanbp	Meanbp	Meanbp
Disbp	Disbp	Disbp
RespRate	TempC	TempC
TempC	RespRate	WBC
PTT	WBC	RespRate
WBC	PTT	PTT
HeartRate	HeartRate	HeartRate
BUN	BUN	SpO2
Potassium	Anion Gap	Anion Gap
SpO2	Bicarbonate	BUN
Anion Gap	SpO2	Bicarbonate
Bicarbonate	Potassium	Potassium

The ranking of features did not change much from considering 8, 6 or 4 hours of clinical data before the patients received vasopressors. So we did not expect a huge change in the results. To prove this

we decided to perform two tests: **Test 4.1 - kNN applied to features individually using a feature selection based on the feature ranking calculated using the six hours/windows before the patients received vasopressors** (Table 4.6). The final prediction is the most frequent prediction among the local predictions; **Test 4.2 - kNN applied to features individually using a feature selection based on the feature ranking calculated using the 4 hours/windows before the patients received vasopressors** (Table 4.6). The final prediction is the most frequent prediction among the local predictions. Figure 4.6 shows the results.

Figure 4.6: Experiment 4 - Results.



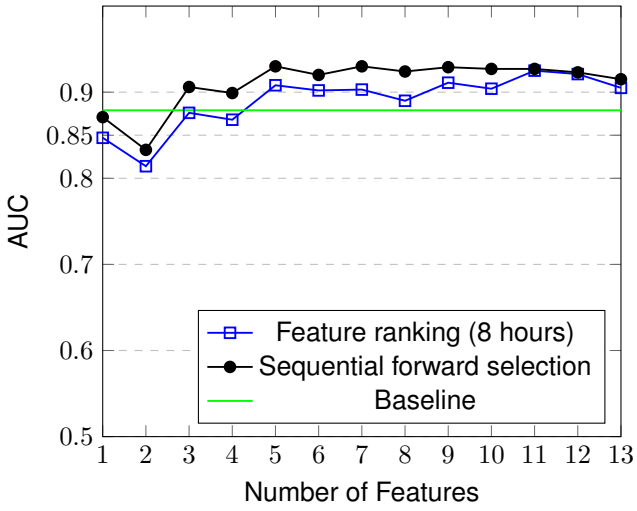
As it was expected the differences between the three feature selection processes tested were almost zero. However, the feature ranking based on the 8 hours before the patients received vasopressors gave slightly better results.

4.4.7 Final Models and Comparison with the State of the Art

The experiments previously done allow us to find the best parameterization for a model based on kNN. For the prediction of vasopressors administration within one hour, the best model is kNN applied to each feature separately (local prediction) and the most frequent prediction among the local predictions is the final prediction (**Model 1**). One goal of this thesis is that the proposed models must use a small set of features in order to be possible for physicians to interpret the predictions. With that in mind, we saw that the sequential forward feature selection and a feature selection based on the feature ranking calculated from using the 8 hours of clinical data before the patients received vasopressors, were the two most effective methods to select this small set of features. Figure 4.7 shows the results of applying **Model 1** using both feature selection procedures.

As we can see in Figure 4.7 the best balance between the number of features used and the resulted

Figure 4.7: Prediction of vasopressors administration within one hour according to two different feature selection process.



AUC was achieved when we are considering 5 features. Table 4.7 summarizes the results.

Table 4.7: Comparison of feature selection processes (1 hour predictions).

Feature selection	Features	AUC
Feature ranking (8 hours)	SysBP, MeanBP, RespRate,DiasBP, Temp	0.908±0,011
Sequential forward selection	SysBP, MeanBP, RespRate, SpO2, Bicarbonate	0.929±0.009
Feature ranking (8 hours)	11 (Table 4.2)	0.925±0,011

Observing Table 4.7 we can see that the difference between the two methods (considering 5 features) is just 0.02, so we think that should be the physicians to choose which 5 features they believe that are more important for the prediction of vasopressors administration.

Since the results for the prediction of vasopressors administration within one hour were so positive, the next natural step was to apply **Model 1** (kNN to each feature individually) for the prediction of vasopressors administration within two hours. We used the two most effective feature selection procedures: the sequential forward feature selection and a feature selection based on the feature ranking calculated from using the 8 hours of clinical data before the patients received vasopressors. Table 4.8 shows the feature order followed by the sequential forward selection method and Figure 4.8 shows the results.

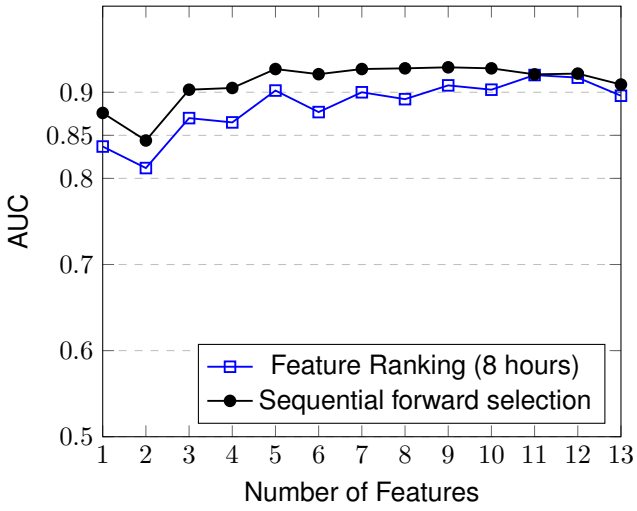
As we can see the results for the prediction of vasopressors administration in the next two hours were very similar with the results achieved for one hour predictions: (i) the sequential forward feature selection returned slightly better results; and (ii) the best balance between the number of features used and the results was achieved when we are considering 5 features. Table 4.9 summarizes the results.

Although the results were very similar between making prediction with one or two hours in advance, there is one difference that deserves attention. The feature sequence used by the sequential forward feature selection was different (Table 4.4 versus Table 4.8). This is one of the biggest disadvantages of

Table 4.8: Best feature order according to feature forward selection for the prediction of vasopressors administration within two hours.

Feature
SpO2
Anion Gap
SysBP
RespRate
MeanBP
Creatinine
HeartRate
TempC
DiasBP
Potassium
PTT
Age
Hematocrit
WBC

Figure 4.8: Prediction of vasopressors administration within two hours according to two different feature selection process.



this method. How are we going to explain to a physician that for one hour predictions the model uses one set of features and for two hour predictions the model uses a different set of features? This type of things just contribute for physicians to feel reluctant in using such model. Thus, we suggest/prefer the feature selection based on the feature ranking, because with this feature selection method we have the exact same model for one and two hours predictions.

The comparison of our results with the state of the art [5, 6, 8, 9] is not straightforward because each work used different filtration criteria, different evaluation procedures and even different versions of MIMIC. However we can see in Table 4.10 that our results are similar to the state of the art, with the advantage that our model uses a smaller number of features.

The proposed model (**Model 1**) is able to provide the necessary context for physicians to accept or reject a prediction. For instance, using **Model 1** with the five most important features according to the

Table 4.9: Comparison of feature selection processes (2 hours predictions).

Feature selection	Features	AUC
Feature ranking (8 hours)	SysBP, MeanBP, RespRate, DiasBP, TempC	0.902±0.010
Sequential forward selection	SysBP, MeanBP, RespRate, SpO2, AnionGap	0.927±0.01
Feature ranking (8 hours)	11 (Table 4.2)	0.920±0.010
Sequential Forward Selection	9 (Table 4.8)	0.929±0.009

Table 4.10: Comparison of the results achieved (considering the feature selection based on the feature ranking and the forward feature selection) with the state of the art. In [5] the number of positive patients (received vasopressors) and in [6] the standard deviation achieved are not clear so they are represented with ?.

Paper	Vaso Patients	AUC	Prediction	Features
Ghassemi et al. [6]	8724	0.820±?	1 hour	29
Proposed model	**3412**	**0.908±0.01 /0.929±0.009**	**1 hours**	**5**
Fialho et al. [8]	1696	0.790±0.02	2 hours	10
Salgado et al. [5]	?	0.850±0.01	2 hours	24
Wu et al. [9]	4331	0.920±0.0016	2 hours	19
Proposed model	**3412**	**0.902±0.01 /0.927±0.01**	**2 hours**	**5**

feature ranking (based on the 8 hours of clinical data before the patients received vasopressors) if we want to predict if a patient p is going to need to receive vasopressors in the next hour, we just need to find the patient stored in the medical repository that had a similar evolution of systolic blood pressure, mean blood pressure, diastolic blood pressure, respiration rate and body temperature. Imagine that for systolic blood pressure, mean blood pressure and diastolic blood pressure, a patient x is the patient who shown the most similar evolution and patient x received vasopressors. For respiration rate the most similar patient is a patient y and the patient y received vasopressors. For body temperature the most similar patient is a patient z and the patient z did not receive vasopressors. In this scenario, the model is going to predict that the patient p is going to need to receive vasopressors in the next hour. Associated with the prediction, we can provide to the physician the following information: (i) patient x evolution of systolic blood pressure, mean blood pressure and diastolic blood pressure, and say that this patient received vasopressors; (ii) patient y evolution of respiration rate, and say that this patient received vasopressors; and (iii) alert that patient p had a similar evolution of body temperature to a patient z who did not receive vasopressors. Figure 4.9 shows the information provided to physicians. We claim that with this additional information based on the past patients, physicians are able to validate the predictions.

The additional information provided is based on the similarity between the patient that we want to predict and the past patients stored in the clinical repository. Saying that, it is important to assess the average difference between the patients that we want to predict and the past patients that the predictions were based on, because if they were very different, the additional information provided may not allow physicians to accept or reject a prediction. One easy way to access the average difference is to calculate, for each feature, the absolute difference between each window value of the patient that we want to predict and the most similar patient found in the clinical repository, divided by the number of windows considered. For instance, Table 4.11 shows the evolution of a patient x that we want to predict and a

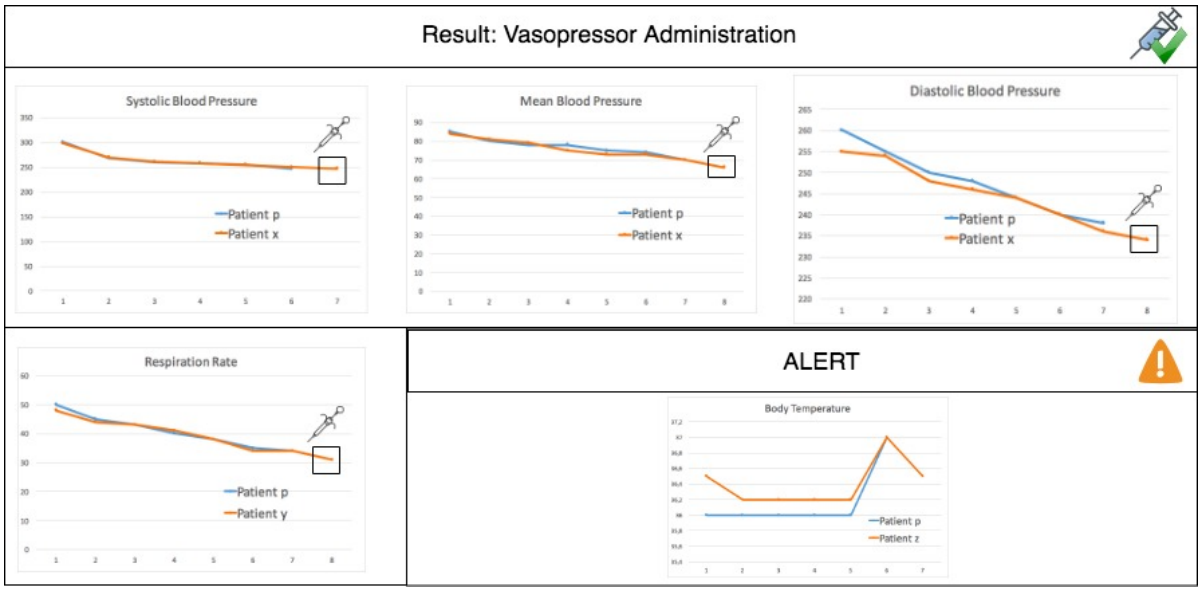


Figure 4.9: Example of context provided to physicians.

patient p that is the most similar patient found for some feature. The average difference between both patients is calculated as follows:

$$AvgDifference = \frac{|0.9 - 0.89| + |0.91 - 0.89| + |0.91 - 0.90|}{3} = 0.13(3) \tag{4.1}$$

Table 4.11: Evolution of patient p that we want to predict and patient p who is the most similar patient for some feature.

Window value patient x	**Window value patient p**
0.9	0.89
0.91	0.89
0.91	0.90

For two hours predictions, **Model 1** with the following two set of features accomplishes all goals of this thesis: *Set 1* - systolic blood pressure, mean blood pressure, diastolic blood pressure, respiration rate and body temperature (feature selection based on the feature ranking); and *Set 2* - systolic blood pressure, mean blood pressure, respiration rate, SpO2 and anion gap (sequential forward selection). We decided to assess the average difference between the patient that we want to predict and the past patients of all features contained in *Set 1* or *Set 2*. Table 4.12 shows the results of the average difference of all features contained in *Set 1* or *Set 2* (following the experimental setup described in Section 4.4.1). As we can see the difference between patients is very small for all features, which shows that this additional information can be useful.

Table 4.12: Average difference and standard deviation between the patient that we want to predict and the past patients that the predictions were based on.

Feature	Avg difference	Std difference
SysBP	0.00822	0.0099
MeanBP	0.0052	0.0067
DiasBP	0.0083	0.0104
RespRate	0.0073	0.0096
TempC	0.0042	0.0064
SpO2	0.0028	0.0088
Anion Gap	0.0011	0.0052

5

Prediction of Mechanical Ventilation

Contents

In this chapter we apply the same methodology to construct a predictive model capable of assisting physicians when they are prescribing the usage of ventilator. We start by producing a predictive model capable of making predictions for the next hour. Like what we did in the previous chapter, we apply the four steps of the methodology. In Section 5.1 we resume the first three steps of the methodology (the actions done were very similar with the actions done for the prediction of vasopressors administration). Then, Section 5.2 describes the construction of the predictive model. In this section we also construct a predictive model capable of making predictions two hours in advance in order to compare the results with the state of the art.

5.1 Overview of the Application of the First Three Steps of the Methodology

In the **data pre-processing** step of the methodology we performed the four tasks in the exact same way that we did for the prediction of vasopressors administration. We did the exact same pre-processing so we could compare the results and conclusions between the two treatments. In the end of this step we had 16 709 patients who did not require mechanical ventilation and 2 921 patients who required mechanical ventilation. A flowchart of the inclusion procedure is depicted in Figure 5.1. The initial features considered were the same features considered for the prediction of vasopressors administration (see Table 4.1).

In the **selection of the patient representation** step we chose the same representation that we did for the prediction of vasopressors: the *closed window represention* with widows of one hour and the average as the aggregation function.

In the **feature ranking** step we analyzed the importance of features in the same two contexts that we did for the prediction of vasopressors administration: (i) considering the 8 hours of clinical data before the patients started using the ventilator; and (ii) considering the 8 hours before and the 4 hours after the patients started using the ventilator. Table 5.1 shows the results of the first context and Table 5.2 shows the results of the second context.

Observing Table 5.1 we conclude that the respiration rate and the SpO2 were the features with higher information gain ratio. The importance revealed by these features is aligned with the main goal of the mechanical ventilation, which is to optimize the oxygenation. Like for the administration of vasopressors, with a simple procedure based on clustering and the information gain ratio we were capable of finding a order of features that seems to be related with the need for mechanical ventilation.

Observing Table 5.2 we see that the feature ranking is different from the the feature ranking resulted from using just the 8 hours of clinical data before the patients started the mechanical ventilation. For instance, the SpO2 considering 8 hours of clinical data is the second most important feature, but con-

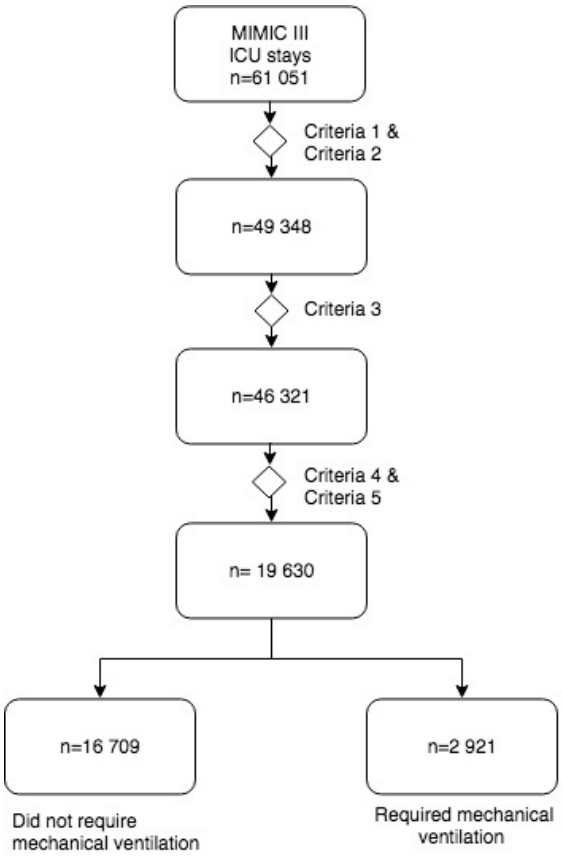


Figure 5.1: Patient selection flowchart

sidering 12 hours of clinical data is the fifth most important feature. Beforehand we did not know which feature ranking was better, but since for the administration of vasopressors the feature ranking based on the 8 hours of clinical data before the patients received the vasopressors was better, we expected that for the mechanical ventilation the feature ranking using the 8 hours of clinical data before the patients started the mechanical ventilation would be also better. Moreover, the feature ranking (Table 5.1) makes more sense for us.

5.2 Construction of Predictive Model

As we did for the prediction of vasopressors administration, the basis of our model was a kNN[1] applied to each feature separately with k=1. To create such model several aspects need to be tested in order to find the best parameterization. We decided to repeat the same four experiments that we did in the construction of the model for the prediction of vasopressors administration: **Experiment 1** - evaluation of the impact of applying kNN to all features together versus applying kNN to each feature

[1] Parameterization: distance function: Euclidean, k=1, R package: [76]. This parameterization is the same for all tests that applied kNN.

Table 5.1: Importance of features considering 8 hours of data.

Feature	*Information Gain Ratio* (Importance)	Groups (K)
RespRate	0,009900579	2
SpO2	0,007697791	3
HeartRate	0,005089688	3
PTT	0,003057256	3
WBC	0,002376722	2
MeanBP	0,002367469	2
TempC	0,002051357	4
Anion Gap	0,001969119	2
Bicarbonate	0,001450316	3
Glucose	0,00123942	3
Platelet	0,001000275	4
Chloride	0,000955757	2
Sodium	0,000797497	4
DiasBP	0,000776144	4
SysBP	0,000629735	4
BUN	0,000620052	5

Table 5.2: Importance of features 12 hours of data.

Feature	*Information Gain Ratio* (Importance)	Groups (K)
HeartRate	0,00481328	4
RespRate	0,004431357	3
PTT	0,003717256	3
WBC	0,002790616	2
SpO2	0,002732344	3
Glucose	0,002009079	3
Anion gap	0,00193474	2
TempC	0,001673917	4
Platelet	0,001354561	3
Bicarbonate	0,001284122	4
MeanBP	0,000767173	2
INR	0,000709565	5
Sodium	0,000609108	4
BUN	0,000601466	5
Potassium	0,000569957	2
Creatinine	0,000501384	4
PT	0,00048381	4
Hematocrit	0,00025722	3

individually (the most frequent prediction among the local predictions is the final prediction); **Experiment 2** - evaluation of the proposed feature selection procedure based on the importance of features versus an automated feature selection procedure (sequential forward selection); **Experiment 3** - evaluation of the impact of using on the feature ranking step different amounts of hours; and **Experiment 4** - evaluation of the impact of weighting the local predictions by the importance of the features. In all experiments we followed the same experimental setup that we followed in the prediction of vasopressors administration. To evaluate the performance of a model, we repeated the following process ten times: (i) construction of a dataset where the number of patients of both types (patients who required mechanical ventilation and patients who did not require mechanical ventilation) was the same; and (ii) application of 10 fold cross validation. Figure 5.2 summarizes the experimental setup.

We started by applying these four experiments in order to find the best model to predict the need

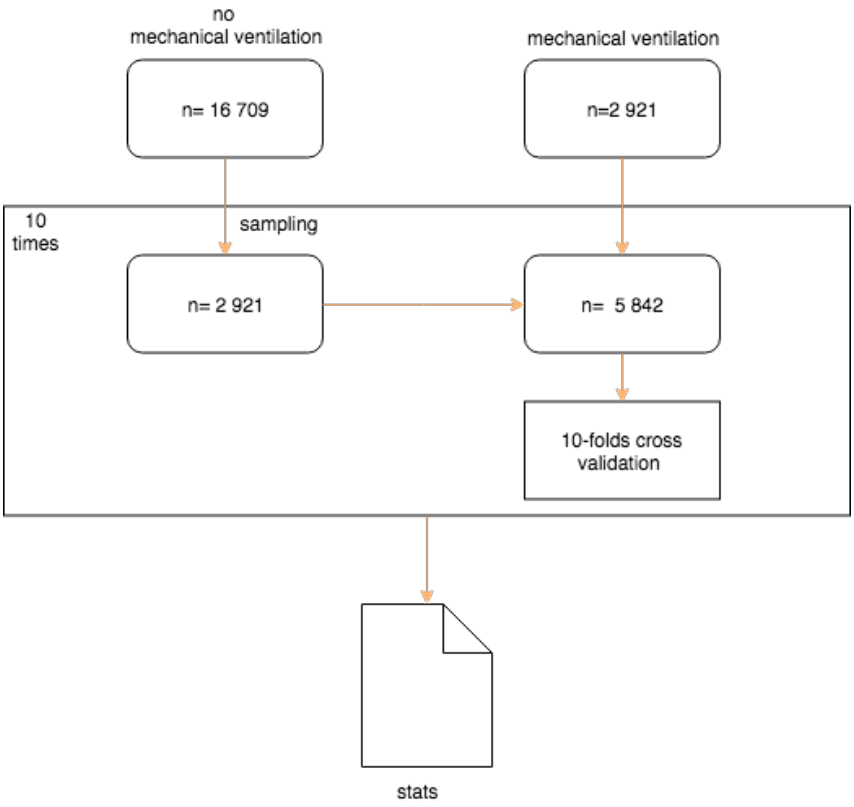


Figure 5.2: Experimental setup flowchart

for mechanical ventilation in the next hour. Then, we apply the same model to predict the need for mechanical ventilation in the next 2 hours.

This section is organized as follows: We start by creating our baseline based on a neural network (Section 5.2.1). Then, we analyze the results of each experiment separately (Section 5.2.2 - Section 5.2.5). And at the end of this section we specify the final model achieved for one and two hours predictions, and we compare the results with the state of the art (Section 5.2.6).

5.2.1 Creation of baseline

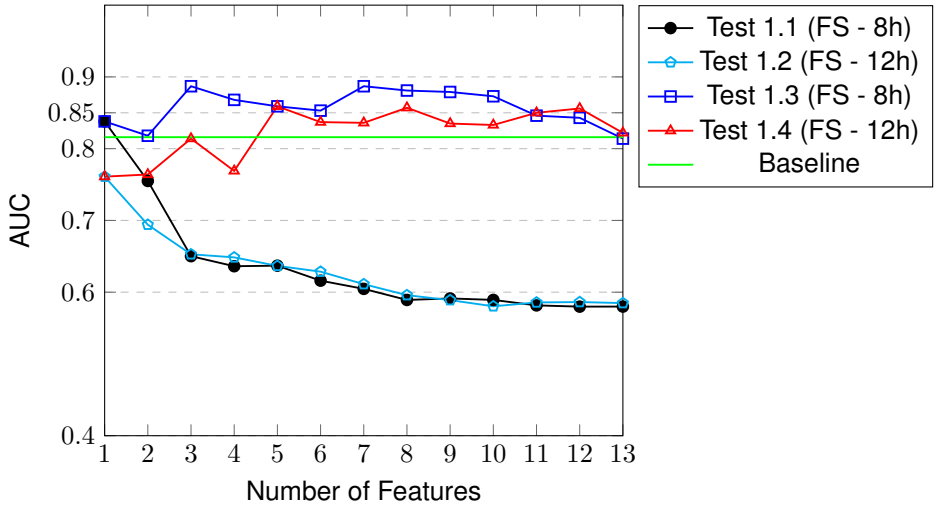
As baseline we applied a neural network[2] to all features together (initial 24 features, see Table 4.1). This test resulted in an AUC of 0.816. This result was lower than the baseline for the prediction of vasopressors administration (0.879), which may indicate that the need for mechanical ventilation is more difficult to predict than the administration of vasopressors, at least using the features that we are considering (Table 4.1).

[2]Parameterization: single hidden layer with 5 Neurons, learning algorithm - BFGS algorithm [77], iterations - 200, decay - 5e-4, rang - 0.1, activation fun - logistic, error - least squares, R package - nnet [76].

5.2.2 Experiment 1 - Impact of Applying kNN to Each Feature Individually

In this experiment we did 4 tests: **Test 1.1 - kNN applied to all features together** using a feature selection based on the feature ranking calculated using the 8 hours of clinical data before the patients started using a ventilator (Table 5.1); **Test 1.2 - kNN applied to all features together** using a feature selection based on the feature ranking calculated using the 8 hours of clinical data before and the 4 hours of clinical data after patients started using a ventilator (Table 5.2); **Test 1.3 - kNN applied to features individually** using a feature selection based on the feature ranking calculated using the 8 hours of clinical data before the patients started using a ventilator (Table 5.1). The final prediction is the most frequent prediction among the local predictions; **Test 1.4 - kNN applied to features individually** using a feature selection based on the feature ranking calculated using the 8 hours before and the 4 hours after the patients started using a ventilator (Table 5.2). The final prediction is the most frequent prediction among the local predictions. Figure 5.3 shows the results of this set of tests.

Figure 5.3: Experiment 1 - Results.



Observing Figure 5.3 we can see some similarities between this experiment and the same experiment performed for the prediction of vasopressors administration: (i) applying a kNN to each feature individually (Test 1.3 and 1.4) outperformed a kNN applied to all features together (Test 1.1 and 1.2); (ii) the feature selection process for kNN applied to all features together (Test 1.1 and Test 1.2) only had a positive impact on the choice of the first feature, from that point forward the results were always worst; (iii) unlike what happened in the prediction of vasopressors administration, with only one feature (Test 1.1 - respiration rate) we beat the baseline, achieving an AUC of 0.838; (iv) the feature selection process based on the feature ranking calculated from using the 8 hours of clinical data before the patients started using a ventilator (Test 1.1 and 1.3) resulted better than the other feature selection process (Test 1.2 and 1.4); and (iv) using only 3 features (respiration rate, SpO2 and heart rate) on a kNN applied to each of

these features individually (Test 1.3) outperformed our baseline, achieving an AUC of 0.887.

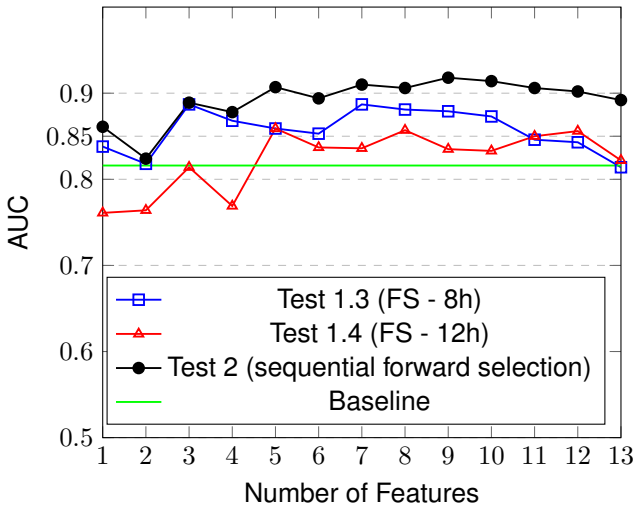
5.2.3 Experiment 2 - Impact of Automated Feature Selection

Like what we did in the prediction of vasopressors administration, we decided to evaluate the impact of using an automated feature selection method, the sequential forward selection (we start with an empty set of features and we add in each step the feature that maximizes the AUC). We did the following test: **Test 2 - apply the sequential forward selection method to a kNN considering each feature individually.** Table 5.3 shows the feature order followed by this method and Figure 5.4 shows the results of this experiment.

Table 5.3: Best feature order according to feature forward selection for the prediction of mechanical ventilation within one hour.

Feature
SpO2
Anion Gap
Resp Rate
MeanBP
Heart Rate
Bicarbonate
DiasBP
TempC
SysBP
Chloride
Age
Platelet
PTT
PT

Figure 5.4: Experiment 2 - Results.



With this experiment we conclude that: (i) the sequential forward selection resulted in a different order of features when compared to both feature rankings calculated in the previous step of the methodology.

The top 5 features according to the sequential forward selection only has 3 features in common with the top 5 features according to both feature rankings; and (ii) until a set of 4 features, the results between using the sequential forward feature selection (Test 2) and the results from using a feature selection based on the feature ranking calculated from using the 8 hours of clinical data before the patients started using a ventilation (Test 1.3) were very similar. However from that point forward the difference between both methods (Test 2 and 1.3) was greater than the difference between both methods in the prediction of vasopressors administration. This can be seen as an evidence that the feature ranking done in the previous step of the methodology was not perfect, maybe the amount of hours used to calculate the feature ranking must be another (in the next experiment we test this hypothesis). In Table 5.4 we can see the differences between the three feature selection processes tested.

Table 5.4: Comparison of the features selection processes tested.

Feature selection	First AUC $>$ Baseline	First AUC $>$ 0.90	Best AUC
Sequential forward selection	0.861$\pm$0.014 (1 features)	0.907$\pm$0.011 (5 features)	0.918$\pm$0.010 (9 features)
Feature ranking (8 hours)	0.838$\pm$0.015 (1 features)	Not Achieved	0.887$\pm$0.012 (3 features)
Feature ranking (12 hours)	0.859$\pm$0.015 (5 features)	Not Achieved	0.859$\pm$0.015 (5 features)

5.2.4 Experiment 3 - Impact of Repeating the Feature Ranking Step

In the previous experiment we saw that the difference between the sequential forward selection and the feature selection based on the feature ranking calculated using the 8 hours of clinical data before the patients started the treatment was higher in the prediction of mechanical ventilation than in the prediction of vasopressors administration. This can be seen as an evidence that the amount of hours used in the feature ranking step was not the best.

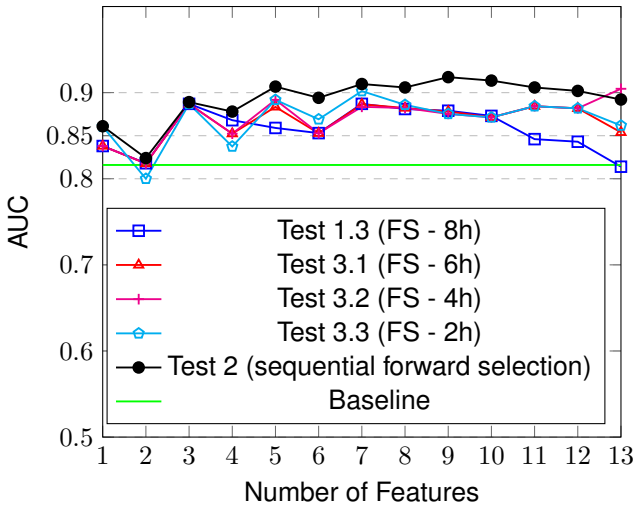
We decided to repeat the feature ranking step considering six, four and two hours of clinical data before the patients started using a ventilator. Table 5.5 shows the order of features resulted from using 6, 4 and 2 hours of clinical data on the feature ranking step.

Table 5.5: Best feature ranking according to feature ranking step using eight, six, four and two hours of clinical data before the patients started the mechanical ventilation.

Features 8 hours	Features 6 hours	Features 4 hours	Features 2 hours
RespRate (0,0099)	RespRate (0,0100)	RespRate (0,0099)	SpO2 (0,0117)
SpO2 (0,0077)	SpO2 (0,0086)	SpO2 (0,0094)	HeartRate (0,0091)
HeartRate (0,0051)	HeartRate (0,0059)	HeartRate (0,0068)	RespRate (0,0086)
PTT (0,0031)	MeanBP (0,0029)	MeanBP (0,0030)	WBC (0,0037)
WBC (0,0024)	PTT (0,0028)	WBC (0,0029)	MeanBP (0,0035)
MeanBP (0,0024)	WBC (0,0026)	PTT (0,0027)	TempC (0,0033)
TempC (0,0021)	TempC (0,0022)	Anion Gap (0,0027)	Anion Gap (0,0030)
Anion Gap (0,0020)	Anion Gap (0,0022)	TempC (0,0026)	Glucose (0,0023)
Bicarbonate (0,0015)	Bicarbonate (0,0017)	Glucose (0,0019)	PTT (0,0022)
Glucose (0,0012)	Glucose (0,0015)	Bicarbonate (0,0014)	Bicarbonate (0,0020)
Platelet (0,0010)	DiasBP (0,0013)	DiasBP (0,0014)	DiasBP (0,0016)
Chloride (0,0010)	Platelet (0,0011)	Platelet (0,0013)	Platelet (0,0014)
Sodium (0,0008)	Chloride (0,0009)	SysBP (0,0010)	PT (0,0013)

We decided to perform three tests: **Test 3.1 - kNN applied to features individually using a feature selection based on the feature ranking calculated using the six hours of clinical data before the patients started using a ventilator**. The final prediction is the most frequent prediction among the local predictions; **Test 3.2 - kNN applied to features individually using a feature selection based on the feature ranking calculated using the 4 hours of clinical data before the patients started using a ventilator**. The final prediction is the most frequent prediction among the local predictions; **Test 3.3 - kNN applied to features individually using a feature selection based on the feature ranking calculated using the 2 hours of clinical data before the patients started using a ventilator**. The final prediction is the most frequent prediction among the local predictions. Figure 5.5 shows the results.

Figure 5.5: Experiment 3 - Results.



With this experiment we conclude that: (i) a feature selection based on the feature ranking calculated using less hours of clinical data before the patients started the mechanical ventilation resulted better. This evidence can mean that this treatment is associated with a sudden health condition change. Table 5.6 shows a comparison between the feature selection processes tested; and (ii) the difference between using the sequential forward selection and the feature selection based on the feature ranking calculated using 4 and 2 hours of clinical data decreased.

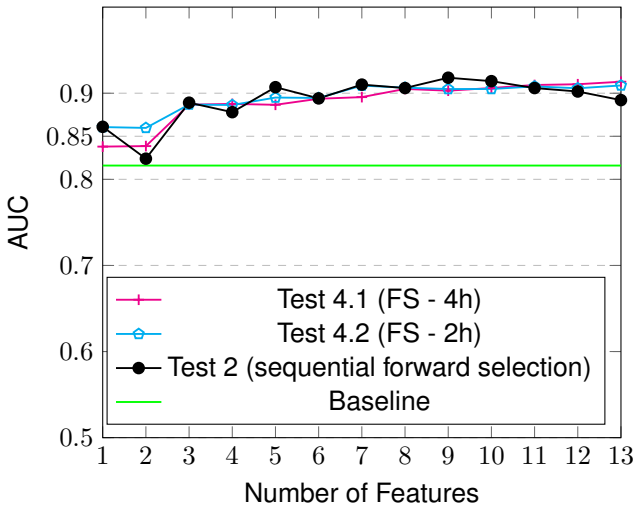
Table 5.6: Comparison of the feature selection processes tested.

Feature selection	First AUC > Baseline	First AUC > 0.90	Best AUC
Sequential Forward Selection	0.861±0.014 (1 features)	0.907±0.011 (5 features)	0.918±0.010 (9 features)
Feature ranking (2 hours)	0.861±0.014 (1 features)	0.902±0.011 (7 features)	0.902±0.011 (7 features)
Feature ranking (4 hours)	0.838±0.015 (1 features)	0.904±0.011 (13 features)	0.904±0.011 (13 features)
Feature ranking (6 hours)	0.838±0.015 (1 features)	Not Achieved	0.887±0.012 (3 features)
Feature ranking (8 hours)	0.838±0.015 (1 features)	Not Achieved	0.887±0.012 (3 features)

5.2.5 Experiment 4 - Impact of Weighting the predictions

In the previous experiment, we saw that the feature ranking based on less hours of clinical data returned better results. So, we used the feature ranking calculated using 4 and 2 hours of clinical data before the patients started the mechanical ventilation to weight the predictions. We performed two tests: **Test 4.1 - kNN applied to each feature individually weighted by the importance of the features (feature ranking 4 hours)**. We also applied a feature selection based on the feature ranking resulted from using the 4 hours of clinical data before the patients started the mechanical ventilation (Table 5.5); and **Test 4.2 - kNN applied to each feature individually weighted by the importance of the features (feature ranking 2 hours)**. We also applied a feature selection based on the feature ranking resulted from using the 2 hours of clinical data before the patients started the mechanical ventilation (Table 5.5). Figure 5.6 shows the results.

Figure 5.6: Experiment 4 - Results.



With this experiment we conclude that: (i) unlike what happened in the prediction of vasopressors administration, weighting the local predictions by their importance resulted better than just derive the final prediction based on the majority of the local predictions. Table 5.7 shows a comparison of the different methods; (ii) the feature selection based on the feature ranking calculated using 2 hours of clinical data achieved a better balance between results (AUC) and the number of features used when compared to the feature selection based on the feature ranking calculated from using 4 hours of clinical data; and (iii) there was almost no difference between the feature selection methods tested.

5.2.6 Final Models and Comparison with the State of the Art

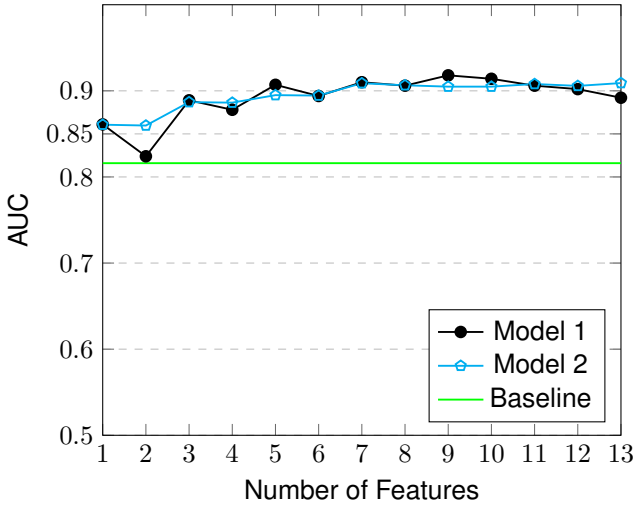
The experiments previously done allow us to find the best parameterization for a model based on kNN. For the prediction of mechanical ventilation within one hour, we found two models that returned

Table 5.7: Comparison of the features selection processes tested and different ways of deriving the final prediction.

feature selection	Final prediction	First AUC $>$ 0.90	Best AUC
Sequential Forward Selection	Majority	0.907±0.011 (5 features)	0.918±0.010 (9 features)
Feature ranking (2 hours)	Weighted	0.909±0.012 (7 features)	0.913±0.013 (15 features)
Feature ranking (4 hours)	Weighted	0.905±0.012 (8 features)	0.913±0.011 (13 features)
Feature ranking (2 hours)	Majority	0.902±0.011 (7 features)	0.902±0.011 (7 features)
Feature ranking (4 hours)	Majority	0.904±0.011 (13 features)	0.904±0.011 (13 features)

similar results: **Model 1** - kNN applied to each feature separately (local prediction) and the most frequent prediction is the final prediction (feature selection - sequential forward selection, Table 5.3); and **Model 2** - kNN applied to each feature separately (local prediction) and the final prediction is the most frequent prediction weighted by the importance of each feature (feature selection - feature ranking calculated from using the 2 hours before the patients started using a ventilator, Table 5.5). Figure 5.7 shows the results of applying both models.

Figure 5.7: Prediction of mechanical ventilation within one hour using two different models.



One of the goals of this thesis is that the proposed models must use a small set of features in order to be possible for physicians interpret the predictions. Observing Figure 5.7, we can see that the best balance between the number of features used and the AUC was achieved when we considered 5 features. Table 5.8 summarizes the results.

Table 5.8: Comparison of two models for the prediction of mechanical ventilation (1 hour predictions).

Model	Features	AUC
Model 1	SpO2, RespRate, MeanBP, HeartRate, Anion Gap	0.907±0,011
Model 2	SpO2, RespRate, MeanBP, HeartRate, WBC	0.895±0.011
Model 1 (best result)	9 (Table 5.3)	0.916±0,012
Model 2 (best result)	14 (Table 5.5 2 hours)	0.910±0,011

As we can see in Table 5.8 the difference between the models (considering 5 features) is just 0.012. Basically, model 1 uses the anion gap and model 2 uses white blood cell count, the remaining 4 features

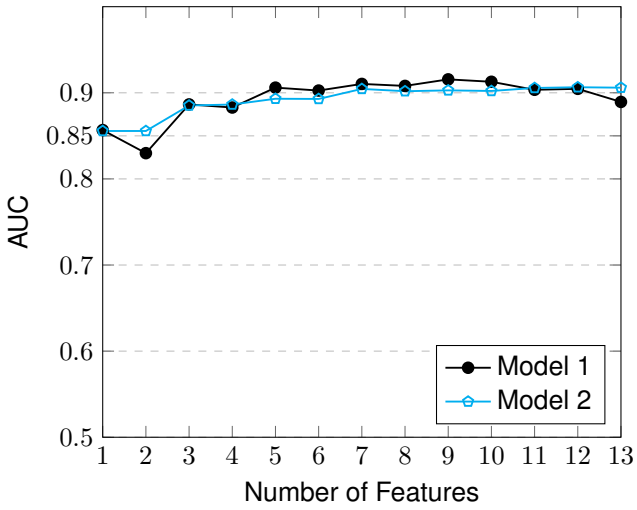
are the same for both models. Saying that, like what happened for the prediction of vasopressors administration, should be the physicians to choose which 5 features they believe are more important for the prediction of mechanical ventilation.

Since the results for the prediction of mechanical ventilation within one hour were so positive, the next natural step was to apply both models (model 1 and model 2) for the prediction of mechanical ventilation within two hours (to do this we discarded in our experimental population the 7 window in all patients). Table 5.9 shows the feature order followed by the sequential forward selection method and Figure 5.8 shows the results.

Table 5.9: Best feature order according to feature forward selection for the prediction of mechanical ventilation within two hour.

Feature
SpO2
Anion Gap
Resp Rate
DiasBP
SysBP
Bicarbonate
Heart Rate
TempC
MeanBP
Creatinine
Age
PTT
PT

Figure 5.8: Prediction of mechanical ventilation within two hours according to two different models. Model 1 follows the order of features present on Table 5.9.



The results for the prediction of mechanical ventilation in the next two hours were very similar with the results achieved for one hour predictions: (i) almost no difference between Model 1 and 2 in terms of results (AUC); and (ii) the best balance between the number of features used and the results was

achieved when we are considering 5 features. Table 5.10 summarizes the results.

Table 5.10: Comparison of two models for the prediction of mechanical ventilation (2 hour predictions).

Model	Features	AUC
Model 1	SpO2, RespRate, Anion Gap, DiasBP, SysBP	0.906±0,011
Model 2	SpO2, RespRate, MeanBP, HeartRate, WBC	0.893±0.011
Model 1 (best result)	9 (Table 5.3)	0.916±0,012
Model 2 (best result)	14 (Table 5.5 2 hours)	0.910±0,011

Like what happened in the prediction of vasopressors administration, the feature sequence used by the sequential forward selection method for one hour prediction was different from the feature sequence used for two hour predictions (Table 5.3 and Table 5.9). This is one of the biggest disadvantages of the sequential forward method.

The comparison of the results obtained with the state of the art [6] is shown in Table 5.11.

Table 5.11: Comparison of the results achieved with the state of the art.

Paper	Number of positives	AUC	Prediction	Features
Ghassemi et al. [6]	17103	0.68±?	1 hour	29
Proposed methodology	**2921**	**0.895±0.011 (Model 2) /0.907±0.011 (Model 1)**	**1 hours**	5
Proposed methodology	**2921**	**0.905±0.014 (Model 2) /0.916±0.012 (Model 1)**	**1 hours**	9
Ghassemi et al. [6]	17103	0.68±?	2 hour	29
Proposed methodology	**2921**	**0.893±0.011 (Model 2) /0.906±0.011 (Model 1)**	**2 hours**	5
Proposed methodology	**2921**	**0.903±0.013 (Model 2) /0.916±0.012 (Model 1)**	**2 hours**	9

Using the proposed models and the five most important features we are able to provide the necessary context for physicians to accept or reject a prediction. For instance using **Model 1**, if we want to predict if a patient p is going to need to use a ventilator in the next hour, we just need to find for each of the 5 features, the patient stored in the medical repository that had a similar evolution. Imagine that for SpO2, anion gap, respiration rate and mean blood pressure, a patient x is the patient with the most similar evolution (for each of the features) and patient x required a ventilator. For heart rate, the most similar patient is a patient y and patient y did not require a ventilator. In this scenario, *model 1* is going to predict that patient p is going to use a ventilator in the next hour. With the prediction of patient p we can provide to the physician the following information: (i) patient x evolution of SpO2, anion gap, respiration rate and mean blood pressure, and say that this patient required a ventilator; and (ii) alert that this patient had a similar evolution of heart rate to a patient y that did not require a ventilator. With this additional information based the on past the patients, physicians can really accept or reject a prediction.

Like what we did for the prediction of vasopressors administration, it is important to assess the difference between the patients that we want to predict and the the most similar patient found in the clinical data for each of the features considered. We assessed the difference for the two best feature sets for the prediction of mechanical ventilation within two hours: *Set 1* - SpO2, respiration rate, anion gap, diastolic blood pressure and systolic blood pressure; and *Set 2* - SpO2, respiration rate, mean blood pressure, heart rate, and white blood cell count. Table 5.12 shows the results of the average difference of

all features contained in *Set 1* or *Set 2* (following the experimental setup described in Section 4.4.1). As we can see the difference between patients is very small for all features, which shows that this additional information can be useful.

Table 5.12: Average difference and standard deviation between the patient that we want to predict and the past patients that the predictions were based on.

Feature	Avg difference	Std difference
WBC	0.0004	0.0035
Anion Gap	0.0012	0.0049
SpO2	0.0027	0.0081
HeartRate	0.0039	0.0052
MeanBP	0.0052	0.0071
DiasBP	0.0059	0.0074
RespRate	0.0070	0.0094
SysBP	0.0085	0.0097

6

Conclusion

Contents

6.1 Summary

Nowadays large amounts of data regarding the patient stays in medical units are stored in digital repositories. For instance, laboratory test results and treatment dates. These repositories have an enormous potential to be used as data source in predictive models that assist physicians when they are prescribing a treatment for a new patient. However the data stored in these clinical repositories are difficult to handle mainly due to high dimensionality and temporal behavior.

Despite these challenges there are different works [5, 6, 8, 9] proposing predictive models for assisting physicians when they making decisions. However, the proposed predictive models do not enable physicians to interpret the predictions. On the one hand, because the state of the art predictive models usually require a large set of features ($>$19), making impossible the interpretation of the predictions and on the other hand because these predictive models do not provide additional information with the predictions so that physicians can take a justified decision of accepting or rejecting the predictions. Moreover, most of the predictive models are generated by complex algorithms, making it difficult to put in practice. These limitations were our motivation to propose predictive models composed by simple methods. The proposed models use 5 features and provide additional information to enable physicians to accept or reject a prediction.

The proposed models in this work followed a methodology that is composed by four steps: (i) data pre-processing; (ii) selection of the patient representation; (iii) feature ranking and (iv) construction of the predictive model. In the first step of the methodology, four important tasks were executed: selection of patient records (based on a patient profile defined), identification of the relevant features, selection of feature measurements and data normalization. In the second step of the methodology, we selected a patient representation to solve the data problems inherent from the fact that we mainly dealt with temporal data - *closed window representation*. In the third step of the methodology, we calculated the importance of each feature for the prediction in consideration. To calculate the importance of features we applied clustering and the information gain ratio. The importance of a feature was the maximum information gain ratio associated to divide the dataset in N clusters considering only the evolution of that feature. The last step of the methodology was the construction of a predictive model whose predictions can be validated by physicians. We used a model based on kNN with k=1 applied to each feature separately. A model based on kNN explores the intuition of similarity between patients, making the model more easily interpretable by physicians. The prediction can be accompanied with the similar patients that the prediction was based on.

For the prediction of vasopressors administration, we constructed a model based on kNN (k=1). Basically, we applied a kNN to each feature separately (local prediction) and the most frequent prediction among the local prediction was the final prediction. For two hour predictions, we used two sets of 5 features: *Set 1* (feature selection based on feature ranking) - systolic blood pressure, mean blood

pressure, respiration rate, diastolic blood pressure and body temperature; and *Set 2* (sequential forward feature selection) - systolic blood pressure, mean blood pressure, respiration rate, SpO2 and anion gap. We achieved AUCs of 0.902 and 0.927 using respectively the *Set 1* and 2.

For the prediction of mechanical ventilation, we reached the two best models based on kNN (k=1): *Model 1* - kNN applied to each feature individually (local predictions) and the final prediction was the most frequent prediction among the local predictions; and *Model 2* - kNN applied to each feature individually (local prediction) and the final prediction was the most frequent prediction weighted by the importance of the features calculated on the feature ranking step. For two hours predictions, *Model 1* with SpO2, respiration rate, anion gap, diastolic blood pressure and systolic blood pressure achieved an AUC of 0.906, and *Model 2* with SpO2, respiration rate, mean blood pressure, heart rate and white blood cell count achieved an AUC of 0.893.

For both treatments we propose models based on simple methods that only require 5 features to achieve results align the state of the art. Moreover, since we are using models based on kNN, associated with the predictions, we can provide the clinical history of past patients that the predictions were based on. By doing this, we accomplished all goals of this thesis.

After all we expect to have demonstrated that using simple and generalizable methods focus on the interpretability of the predictions, we can also achieve very competitive results and have models that can be seen as a useful second opinion by physicians.

6.2 Limitations and Future Work

There are many interesting paths that can be further explored: (i) incorporate a step in the methodology dedicated to communicate the predictions and respective context to the physicians; (ii) assess the real impact of the window size on the patient representation and the impact of the aggregation functions used to represent the windows; and (iii) evaluate the impact of using other clustering algorithms (e.g., hierarchical) on the feature ranking step.

This thesis has three main limitations: (i) we only used one clinical repository in the validation of our models; (ii) static data (e.g., gender) is not fully explored by kNN applied to each feature separately; and (iii) we calculated the AUC based on the confusion matrix, instead of varying a decision threshold. In future work we should use other evaluation metrics such as precision and recall.

Bibliography

[1] C. C. Aggarwal and C. K. Reddy, *DATA Custering Algorithms and Applications*. Chapman & Hall/CRC, 2013.

[2] S. B. Kotsiantis, "Supervised Machine Learning: A Review of Classification Techniques," *Informatica*, vol. 31, pp. 249–268, 2007.

[3] H. Ding, G. Trajcevski, P. Scheuermann, X. Wang, and E. J. Keogh, "Querying and mining of time series data: experimental comparison of representations and distance measures," *Proc. of the VLDB Endowment*, vol. 1, no. 2, pp. 1542–1552, 2008.

[4] A. E. W. Johnson, T. J. Pollard, L. Shen, L.-w. H. Lehman, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. A. Celi, and R. G. Mark, "Data Descriptor : MIMIC-III , a freely accessible critical care database," *Scientific Data*, vol. 3, pp. 1–9, 2016.

[5] C. M. Salgado, S. M. Vieira, L. F. Mendonça, S. Finkelstein, and J. M. Sousa, "Ensemble fuzzy models in personalized medicine: Application to vasopressors administration," *Engineering Applications of Artificial Intelligence*, vol. 49, pp. 141–148, 2016.

[6] M. Ghassemi, M. Wu, M. C. Hughes, P. Szolovits, and F. Doshi-Velez, "Predicting intervention onset in the ICU with switching state space models." *AMIA Joint Summits on Translational Science proceedings. AMIA Joint Summits on Translational Science*, vol. 2017, pp. 82–91, 2017.

[7] W. Raghupathi and V. Raghupathi, "Big data analytics in healthcare: promise and potential," *Health Information Science and Systems*, vol. 2, no. 1, p. 10, 2014.

[8] A. S. Fialho, L. A. Celi, F. Cismondi, S. M. Vieira, S. R. Reti, J. M. Sousa, and S. N. Finkelstein, "Disease-based modeling to predict fluid response in intensive care units," *Methods of Information in Medicine*, vol. 52, no. 6, pp. 494–502, 2013.

[9] M. Wu, M. Ghassemi, M. Feng, L. A. Celi, P. Szolovits, and F. Doshi-Velez, "Understanding vasopressor intervention and weaning: Risk prediction in a public heterogeneous clinical time series

database," *Journal of the American Medical Informatics Association*, vol. 24, no. 3, pp. 488–495, 2017.

[10] A. E. W. Johnson, D. J. Stone, L. A. Celi, and T. J. Pollard, "The MIMIC Code Repository : enabling reproducibility in critical care research," *Journal of the American Medical Informatics Association : JAMIA*, vol. 25, no. September 2017, pp. 32–39, 2018.

[11] T. Takagi and M. Sugeno, "Fuzzy identification of system and its applications to modeling and control," in *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-15, no. 1, 1985, pp. 116–132.

[12] L. W. H. Lehman, R. P. Adams, L. Mayaud, G. B. Moody, A. Malhotra, R. G. Mark, and S. Nemati, "A physiological time series dynamics-based approach to patient monitoring and outcome prediction," *IEEE Journal of Biomedical and Health Informatics*, vol. 19, no. 3, pp. 1068–1076, 2015.

[13] F. Costa, H. Galhardas, M. J. Fonseca, and J. Pereira, "Metodologia para criação de modelos preditivos de suporte à decisão médica," in *Inforum 2018 - Simpósio de Informática*, 2018.

[14] H. Jiawei, M. Kamber, J. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques*. Morgan Kaufmann, 2012.

[15] J. B. MacQueen, "Some Methods for classification and Analysis of Multivariate Observations," in *Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability*, vol. 1, 1967, pp. 281–297.

[16] J. C. Dunn, "A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters," *Journal of Cybernetics*, vol. 3, no. 3, pp. 32–57, 1973.

[17] J. C. Bezdek, R. Ehrlich, and W. Full, "FCM: The fuzzy c-means clustering algorithm," *Computers & Geosciences*, vol. 10, no. 2-3, pp. 191–203, 1984.

[18] I. S. Dhillon, "Kernel k-means , Spectral Clustering and Normalized Cuts," in *Proceedings of the 2004 ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '04*, 2004, pp. 551–556.

[19] H. S. Park and C. H. Jun, "A simple and fast algorithm for K-medoids clustering," *Expert Systems with Applications*, vol. 36, no. 2, pp. 3336–3341, 2009.

[20] R. Tibshirani, G. Walther, and T. Hastie, "Estimating the number of clusters in a data set via the gap statistic," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 63, no. 2, pp. 411–423, 2001.

[21] M. Meilă, “The uniqueness of a good optimum for K-means,” *Proceedings of the 23rd international conference on Machine learning - ICML ’06*, pp. 625–632, 2006.

[22] J. G. Wilpon and L. R. Rabiner, “Modified k-means clustering algorithm for use in isolated word recognition,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-33, no. 3, pp. 587–594, 1985.

[23] M. K. Pakhira, “A Linear Time-Complexity k-Means Algorithm Using Cluster Shifting,” in *Sixth International Conference on Computational Intelligence and Communication Networks*, no. November, 2014, pp. 1049–1053.

[24] J. Nayak, B. Naik, and H. S. Behera, “Fuzzy C-means (FCM) clustering algorithm: A decade review from 2000 to 2014,” in *Smart Innovation, Systems and Technologies*, vol. 32, 2015, pp. 133–149.

[25] C. Ding and X. H. X. He, “Cluster merging and splitting in hierarchical clustering algorithms,” in *IEEE International Conference on Data Mining*, 2002, pp. 1–8.

[26] A. K. Jain, M. N. Murty, and P. J. Flynn, “Data clustering: a review,” *ACM Computing Surveys*, vol. 31, no. 3, pp. 264–323, 1999.

[27] M. Steinbach, “A Comparison of Document Clustering Techniques,” in *KDD Workshop on Text Mining*, 2000, pp. 1–2.

[28] H. P. Kriegel, P. Kröger, J. Sander, and A. Zimek, “Density-based clustering,” *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 1, no. 3, pp. 231–240, 2011.

[29] M. Ester, H. P. Kriegel, J. Sander, and X. Xu, “A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise,” *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining*, pp. 226–231, 1996.

[30] M. Ankerst, M. M. Breunig, and H.-p. Kriegel, “OPTICS : Ordering Points To Identify the Clustering Structure,” in *SIGMOD Philadelphia*, 1999, pp. 49–60.

[31] T. Warren Liao, “Clustering of time series data - A survey,” *Pattern Recognition*, vol. 38, no. 11, pp. 1857–1874, 2005.

[32] E. Schikuta, “Grid Clustering: A Fast Hierarchical CLustering Methid for Very Large Data Set,” in *Proceedings of the 13th International Conference on Pattern Recognition*, 1993, pp. 101–105.

[33] W. Wang, J. Yang, and R. Muntz, “STING : A Statistical Information Grid Approach to Spatial Data Mining,” in *VLDB Athens*, 1997, pp. 186–195.

[34] G. Sheikholeslami, S. Chatterjee, and A. Zhang, “Wavecluster: A multi-resolution clustering approach for very large spatial databases,” *Proceedings of the International Conference on Very Large Data Bases*, no. 24, pp. 428–439, 1998.

[35] G. Gan, C. Ma, and J. Wu, *Data Clustering: Theory, Algorithms, and Applications*. ASASIAM Series on Statistics and Applied Probability, 2007, vol. 20.

[36] P. Grabusts and A. Borisov, “Using grid-clustering methods in data classification,” in *International Conference on Parallel Computing in Electrical Engineering, PARELEC*, 2002, pp. 425–426.

[37] N. P. Lin, C.-i. Chang, and C.-l. Pan, “An Adaptable Deflect and Conquer Clustering Algorithm,” in *Proceedings of the 6th International Conference on Applied Computer Science—Volume 6, ACOS’07*, 2007, pp. 155–159.

[38] A. Hinneburg and D. a. Keim, “Optimal Grid-Clustering: Towards Breaking the Curse of Dimensionality in High-Dimensional Clustering,” *International Conference on Very Large Databases (VLDB)*, pp. 506–517, 1999.

[39] C. R. Shalizi, “Advanced data analysis from an elementary point of view,” *Book Manuscript*, p. 801, 2017.

[40] A. P. Dempster, N. M. Laird, and D. B. Rubin, “Maximum Likelihood from Incomplete Data via the EM Algorithm,” *Journal of the Royal Statistical Society. Series B*, vol. 39, no. 1, pp. 1–38, 1977.

[41] C. Couvreur, “The EM Algorithm: A Guided Tour,” in *Computer Intensive Methods in Control and Signal Processing*, 1997, pp. 209–222.

[42] J. E. Gentle, G. J. McLachlan, and T. Krishnan, “The EM Algorithm and Extensions.” *Biometrics*, vol. 54, no. 1, p. 395, 1998.

[43] S. C. Madeira and A. L. Oliveira, “Biclustering algorithms for biological data analysis: A survey,” *IEEE/ACM Trans. Comput. Biol. Bioinforma.*, vol. 1, no. 1, pp. 24–45, 2004.

[44] Y. Cheng and G. M. Church, “Biclustering of expression data.” *International Conference on Intelligent Systems for Molecular Biology*, vol. 8, pp. 93–103, 2000.

[45] L. Ji and K. L. Tan, “Mining gene expression data for positive and negative co-regulated gene clusters,” *Bioinformatics*, vol. 20, no. 16, pp. 2711–2718, 2004.

[46] O. Odibat and C. K. Reddy, “A generalized framework for mining arbitrarily positioned overlapping co-clusters,” *Proceedings of the 11th SIAM International Conference on Data Mining, SDM 2011*, pp. 343–354, 2011.

[47] J. A. Hartigan, “Direct clustering of a data matrix,” *Journal of the American Statistical Association*, vol. 67, no. 337, pp. 123–129, 1972.

[48] J. Yang, H. Wang, W. Wang, and P. Yu, “Enhanced Biclustering on Experssion Data,” *Proceedings of the IEEE Symposium on Bioinformatics and Bioengineering, BIBE, Bethesda, MD, USA*, pp. 321–327, 2003.

[49] E. Segal, B. Taskar, a. Gasch, N. Friedman, and D. Koller, “Rich probabilistic models for gene expression.” *Bioinformatics (Oxford, England)*, vol. 17 Suppl 1, no. 1, pp. S243–S252, 2001.

[50] E. Keogh and C. A. Ratanamahatana, “Exact indexing of dynamic time warping,” *Knowledge and Information Systems*, vol. 7, no. 3, pp. 358–386, 2005.

[51] G. Kollios, M. Vlachos, and D. Gunopulos, “Trajectories, Discovering Similar,” in *Proceedings of the 18th Internacional Conference on Data Engineering*, 2002, pp. 1–8.

[52] C. Guo, H. Jia, and N. Zhang, “Time series clustering based on ICA for stock data analysis,” in *International Conference on Wireless Communications, Networking and Mobile Computing, WiCOM*, 2008.

[53] W. Meesrikamolkul, V. Niennattrakul, and C. A. Ratanamahatana, “Shape-based clustering for time series data,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 7301 LNAI, no. PART 1, 2012, pp. 530–541.

[54] S. Chandrakala and C. C. Sekhar, “A density based method for multivariate time series clustering in kernel feature space,” *IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, 2008.

[55] S. Aghabozorgi, A. Seyed Shirkhorshidi, and T. Ying Wah, “Time-series clustering - A decade review,” *Information Systems*, vol. 53, pp. 16–38, 2015.

[56] B. Andreopoulos, A. An, X. Wang, and M. Schroeder, “A roadmap of clustering algorithms: Finding a match for a biomedical application,” pp. 297–314, 2009.

[57] Y. Xue, M. Zhang, Z. Liao, M. Li, J. Luo, and X. Hu, “A contiguous column coherent evolution biclustering algorithm for time-series gene expression data,” *International Journal of Machine Learning and Cybernetics*, 2016.

[58] S. C. Madeira and A. L. Oliveira, “An efficient biclustering algorithm for finding genes with similar patterns in time-series expression data,” *Proc. of the 5th Asia Pacific Bioinformatics Conference, Series in Advances in Bioinformatics and Computational Biology. Volume 5.Imperial College Press*, pp. 67–80, 2007.

[59] T. Cover and P. Hart, “Nearest neighbor pattern classification,” *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, 1967.

[60] S. A. Dudani, “The Distance-Weighted k-Nearest-Neighbor Rule,” *IEEE Transactions on Systems, Man and Cybernetics*, vol. SMC-6, no. 4, pp. 325–327, 1976.

[61] G. Guo, H. Wang, D. Bell, Y. Bi, and K. Greer, “KNN model-based approach in classification,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 2888, pp. 986–996, 2003.

[62] A. McCallum and K. Nigam, “A Comparison of Event Models for Naive Bayes Text Classification,” *AAAI/ICML-98 Workshop on Learning for Text Categorization*, pp. 41–48, 1998.

[63] T. M. Mitchell, *Machine Learning*. McGraw-Hill, 1997.

[64] T. N. Phyu, “Survey of Classification Techniques in Data Mining,” *Proceedings of the International MultiConference of Engineers and Computer Scientists 2009 Vol I*, vol. I, 2009.

[65] J. R. Quinlan, “Induction of Decision Trees,” *Machine Learning*, vol. 1, no. 1, pp. 81–106, 1986.

[66] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees*, 1st ed., New York, 1984, vol. 19.

[67] J. R. Quinlan, *C4.5: Programs for Machine Learning*, San Mateo California, 1992, vol. 1, no. 3.

[68] J. Quinlan, *Discovering Rules by Induction from large collections of examples*. Edinburgh University Press, 1979.

[69] V. N. Vapnik, *The Nature of Statistical Learning Theory*. Springer, 1995, vol. 8.

[70] C. J. C. Burges, “A Tutorial on Support Vector Machines for Pattern Recognition,” *Data mining and knowledge discovery*, vol. 2, no. 2, pp. 121–167, 1998.

[71] B. W. White and F. Rosenblatt, “Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms,” *The American Journal of Psychology*, vol. 76, no. 4, p. 705, 1963.

[72] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, *Learining Internal Representations by Error Propagation*. MIT Press, 1986, vol. 1.

[73] L. Breiman, “Random Forests,” *Machine Learning*, vol. 45, no. 5, pp. 1–35, 1999.

[74] X. Xi and C. A. Ratanamahatana, “Fast Time Series Classification Using Numerosity Reduction,” in *23rd international conference on Machine learning*, 2006, pp. 1033–1040.

[75] A. E. Johnson, T. J. Pollard, L. Shen, L. W. H. Lehman, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. Anthony Celi, and R. G. Mark, “MIMIC-III, a freely accessible critical care database,” *Scientific Data*, vol. 3, 2016.

[76] W. N. Venables and B. D. Ripley, “Statistics Complements to Modern Applied Statistics with S,” in *Modern Applied Statistics with S*. Springer, 2002, p. 48.

[77] R. Fletcher, “Practical Methods of Optimization,” *John Wiley & Sons*, vol. 53, p. 456, 2013.