Hypotheses engine (HypE): exploring structured biomedical datasets in search for predictive patterns

Beatriz Neves Mano

School of Science

Thesis submitted for examination for the degree of Master of Science in Technology.
Espoo 15.11.2018

Supervisor

Prof. Samuel Kaski

Advisors

Prof. Hiroshi Mamitsuka

Dr. Henrik Edgren

Aalto University
School of Science

Aalto University
School of Science

Aalto University, P.O. BOX 11000, 00076 AALTO
www.aalto.fi
Abstract of the master's thesis

Author Beatriz Neves Mano

Title Hypotheses engine (HypE): exploring structured biomedical datasets in search for predictive patterns

Degree programme Computer, Communication and Information Sciences

Major Machine Learning and Data Mining **Code of major** SCI3044

Supervisor Prof. Samuel Kaski

Advisors Prof. Hiroshi Mamitsuka, Dr. Henrik Edgren

Date 15.11.2018 **Number of pages** 62 **Language** English

Abstract

Nowadays, healthcare facilities constantly collect an immense amount of data as part of their daily-management systems, which include diverse type of information, such as patient admission details, drugs administered or clinical examinations' results.

Even though medical research has been traditionally condition-oriented, researchers oftentimes use similar analysis methodologies, with very little context customization, making them computationally redundant.

This project proposes an analysis pipeline capable of automatically mine big and diverse biomedical datasets, and identify potentially interesting patterns in the data, despite of the medical conditions the data might relate to. Such system is called an hypotheses engine, as its purpose is to output patterns that seem to be medically predictive, which we call hypotheses.

HypE's novelty is two-fold: on one hand, a tailored data processing method was developed for analyzing inconsistent and chaotic temporal data (i.e. a patient has laboratory measurements, that usually are only partially repeated over time); and on the other hand, the hypotheses found are to be outputted in a physician-friendly way, to allow fast understanding of the patterns found, in case medical intervention is recommended.

Given HypE's functionality, results cannot be straightforwardly classified as good or bad, as certain data subsets might actually not contain any patterns, at all. However, methodologically, it is to expect that some hypotheses found will be known medical patterns. Thus, HypE's outputs are presented and discussed on a high level, considering no manual check for their medical validity was performed by medical experts. The prototype implemented was ran on MIMIC-III data and the results exceeded the initial expectations as they did include common medical scenarios.

Keywords data mining, electronic health records, temporal data, machine learning, linear regression , support vector machines, decision trees

Acknowledgments

First and foremost, I would like to thank my supervisor Prof. Samuel Kaski from School of Science, Department of Computer Science at Aalto University, for his expertise and availability, making it extremely easy for me to contact him, whenever I needed him.

A very special gratitude to my advisor, Prof. Hiroshi Mamitsuka (Kyoto University / Aalto University), who was always available for whatever I needed, consistently allowing this thesis to be my own work, but guiding me in the right direction whenever required.

I would like to add a special mention to Medisapiens, in general, and my advisor Dr. Henrik Edgren, in particular. Dr. Henrik Edgren has helped me immensely, always with so much patience and enlightenment. His support was essential, not only in finding an interesting topic, but also in making the whole research project extremely enjoyable. I am truly thankful to Medisapiens for giving me the opportunity to develop my thesis there, based on real-world scenarios, which have undeniably given me a sense of purpose during the entire project.

Finally, I must express my profound gratitude to my parents, without whom I would have never had the opportunity to study abroad, or at all. They have given me endless support and encouragement throughout my life, in addition to having invested so much in my education, for which I am truly thankful. Last, but not least, I would like to thank my friends, who are, and have been, an essential emotional support, specially during my time abroad.

Thank you all.

Otaniemi, 15.11.2018

Beatriz Neves Mano

Contents

Abbreviations

AI	Artificial intelligence
ANN	Artificial Neural Networks
AUC	Area Under the Curve
BMI	Brain Function Mapping
BN	Bayesian Networks
CAD	Computer-aided Diagnosis
CART	Classification and Regression Trees
CDF	Cumulative Distribution Function
CDSS	Clinical Decision Support System
CNN	Convolutional Neural Networks
CV	Cross-validation
DNA	Deoxyribonucleic Acid
DOD	Date of Death
DT	Decision Tree
EHR	Electronic Health Records
FP	False Positives
HIT	Health Informatics Technology
HypE	Hypothesis Extraction Engine
ICA	Independent Component Analysis
ICD-9	International Classification of Diseases (version 9)
ICU	Intensive Care Unit
KS	Kolmogorov-Smirnov
LOOCV	Leave One-out Cross-validation
MIMIC-III	Medical Information Mart of Intensive Care (version 3)
ML	Machine Learning
PC	Principal Component
RBF	Radial Basis Function
RNA	Ribonucleic Acid
ROC	Receiver Operating Characteristic
SL	Supervised Learning
SVM	Support Vector Machines
TP	True Positives

1 Introduction

1.1 Motivation / Health Informatics

Nowadays, healthcare facilities constantly collect an immense amount of data as part of their daily-management systems. These data are extremely diverse, and may include patient admission details, drugs administered, laboratory results, among other type of information.

Computational resources have greatly upgraded, and artificial intelligence is now living a golden era, being applied to real-world problems like never before, along with biomedical research that has accordingly been developing heavier data-driven methods.

Healthcare is, not only about diagnosing and treating illnesses, but also about preventing them, and biomedical research has been increasingly focusing on the latter, by crossing knowledge from different fields of science, like genetics and genomics, with computational resources and techniques developed by applied sciences, such as bioinformatics. The simultaneous advances of genomic technologies have lead to an enlargement of biological data available for research, which has surely benefited the findings.

Accessible biological data (DNA and RNA sequencing data, mass spectrometry data, etc) combined with computational power, have, to an certain extent, trivialized massive data analysis, which seems to be progressively making feasible the long-term belief that medicine is moving towards personalized care [1]. Personalized medicine (PM) has been frequently described as the tailoring of treatments to the individual patient, but it is a slightly more extensive concept, consisting on the customization of medical decisions based on the patient's genetic profile [2]. For instance, numerous cancer researches have taken the PM approach, using genomic data to either diagnose or predict the occurrence of cancer on potential patients [3].

Machine Learning (ML) has greatly contributed for the progress of clinical medicine, as it has been developing methods that aim at improving the extraction of knowledge from clinical data available. In fact, ML was initially mainly focused in disease diagnostics, namely of several cancer types [4], and its focus was later broaden to disease prediction and prognostics, which is the scope of this thesis.

Health Informatics Technology (HIT) refers to the use of technology to healthcare, seeking to improve both private and public healthcare, as well as biomedical research [5]. An example of HIT are clinical decision support systems (CDSS), which analyze clinical data and subsequently provide users with insights regarding it [6].

Earlier CDSS, and still many of today's, were designed for some specific context, where actual medical knowledge was intentionally incorporated within the system structure [7]. However, considering the significant increase of clinical data being deliberately collected by biomedical research and healthcare facilities, more recent CDSS have been developed exclusively upon ML methods, that automatically search for patterns in clinical data. This work precisely concerns the development of an automatic analyzer tool for clinical data.

1.2 Research proposal

Although modern medicine repeatedly manages to delay death and making conditions that were once fatal no longer deadly, there are still an infinite amount of fatal and non-fatal epidemics that affect millions of people, such as cancer or degenerative diseases.

The most common approach to clinical studies is condition-based, by limiting the analysis to a predefined medical condition of interest, for example breast cancer or Alzheimer's disease. However, these researches often implement the same analysis' methodology, with very little context customization, making them, computation wise, essentially redundant.

Naturally, health care facilities, such as hospitals, have patients suffering from diverse disorders, and such diversity is as inspiring as it is challenging, from the research point of view. Inspiring because its variety may provide, for instance, new insights on correlations across different conditions; challenging because clinical data is undeniably complicated to work with, specially without deliberately curating it for a certain purpose.

The aim of this thesis is to develop an illness-independent system that finds potentially interesting medical phenomena in very large biomedical datasets. This system draws condition-specific scenarios, which are basically nested identified patterns, that are highly suggestive in identifying patients' risk of developing complications. More specifically, these scenarios, exclusively generated from laboratory measurements, explicitly indicate which variables seem of great importance in predicting the patient outcome, such as death, the development of new condition, among other possibilities. Thus, rather than solely pursuing the maximal predictive power of extracted models, the goal of this system is to unmask understanding of the patient state, by alerting physicians' of the likelihood of undesired outcomes.

This work's contribution is twofold: first, it presents a novel pre-processing technique to extract relevant data from time-sequential chaotic clinical data, and second, the system's output is intended to be physician-friendly, i.e., medical specialists are presumed to rapidly understand the scenarios being presented, and, if necessary, intervene accordingly. Figure 1 shows the entire system pipeline and identifies the mentioned contributions.

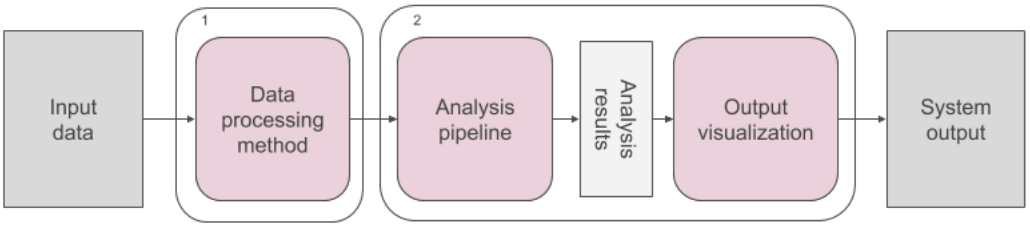

Figure 1: High level illustration of the entire system developed, from input to output, with the work's contributions highlighted.

This document starts with a background Chapter, where the essential compu-

tational concepts needed to understand the work are presented. The related work Chapter then proceeds to summarize the biomedical research that has been developed following the machine learning approach. Chapter 4 describes the data that was used in this work, as well as the pre-processing operations that were done before the analysis itself. Chapter 5 details the entire conceptualized system and its architecture, from the feature engineering different steps, to the output generation. Chapter 6 then proceeds to show some results achieved by the prototype. Chapter 7 discusses the results, while summarizing the entire prototype architecture, as well as all implementation choices. And finally, Chapter 8 concludes this work.

2 Computational Background

This section is mainly dedicated to those who are not familiar with what machine learning (ML) is. It starts with a brief introduction of the concept and the types of ML used in the biomedical domain. It finalizes with a more detailed explanation of supervised learning and the methods used in this work.

2.1 From Artificial Intelligence to Machine Learning

Artificial intelligence (AI) is an extensive interdisciplinary field that focuses on understanding and building intelligence entities. This purposely vague definition was given by Stuart Russell and Peter Norvig, in what is considered to be the bible book of modern AI [8] , and its ambiguity is so to include different approaches to different problems, from the theoretical study of what intelligence actually is, to the study of how to synthetically build entities that exhibit such "intelligent" behavior.

In computer science, AI can be described as the science and engineering of making intelligent machines [9], which encompasses the process of both studying and designing intelligent agents. Intelligent agents are systems that display properties like autonomy, rationality and reactivity, as an agent perceives its environment, acts accordingly in order to maximize its chance of achieving its goals, and does it so without any direct human intervention. Additionally, agents may possess properties like pro-activeness and social ability, with agents behavior being primarily goal-oriented and having the ability to interact, or even cooperate, with other agents [10]. Computationally, AI describes the building of systems that act rationally, not humanely. This distinction is important, because a system acting humanely would be indistinguishable from an actual human being, from a cognitive perspective, while a rational system just means that the system takes the best action to achieve a certain goal. A simple example of an intelligent agent would be a software-based robot that vacuums the floor whenever it finds it dirty.

Because intelligence is such a complex concept, AI has broken ground to many branches of research that focus on different matters, namely knowledge representation, natural language processing, computer vision, among many others.

Machine learning (ML) has been one of the fastest growing branches of AI. Simplistically, ML can be defined as the development of systems that have the ability to extract knowledge from data, i.e., systems that learn and, ideally, such intelligence is fairly capable of predicting future occurrences of phenomena that have already happened before. Naturally, different authors define ML differently, some more mathematically like Alpaydin [11] or more abstractly like Murphy [12], but all of them share the underlying notation of using past data to foresee patterns.

The ways in which the systems themselves learn are closely related to other fields of science, namely statistical analysis, pattern recognition or algorithmic optimization. ML's main task is to infer from examples, with such inferences being acknowledged over the identification of patterns in the data. In order to take the most advantage of available computational power, this process is ideally done through well-designed and efficient algorithms.

2.2 Machine Learning

As with most fields of science, different problems motivate the search for different solutions, which eventually are embodied into different applications. ML is no different, and health informatics domain mainly adopts mechanisms of supervised and unsupervised learning.

Supervised learning consists of using labeled data to infer a certain outcome [13]. The idea is that data tends to be identical within use-cases and hence, having labeled examples should make the process of extracting label-based patterns clearer, which should theoretically lead to models that better identify that certain behaviour. For example, if a model is built to estimate the risk of someone having a heart disease [14], given information like the person's age, amount of times they reported chest pain and electrocardiograph results, it should be fairly straightforward to assess, in the future, if some other person is at risk or not of suffering a heart attack, as Figure 2 illustrates; however, if we give this same model the amount of hours of exercise, or some other random information as the model's attributes, the model itself will not know the difference of the given data's meaning and thus its output will have no interpretive meaning.

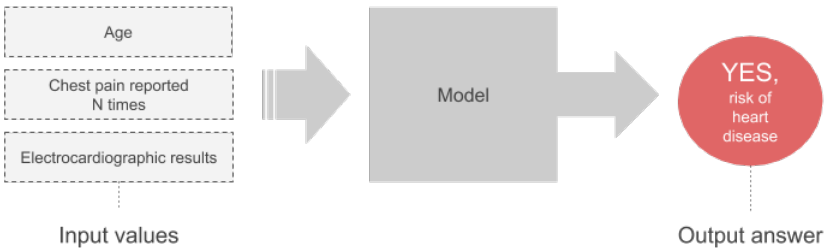

Figure 2: Example demonstrating how predictive models work. In this case, the input values belonging to a patient are fed into the model, which outputs if the patient is at risk of having a heart disease or not. Based on [14].

Unsupervised learning, on the other hand, does not involve label information. Its goal is mostly about data discovery, without any previous knowledge of trends or similarities in the data. As such, unsupervised learning is often associated with data exploration and the quest for data structure insight, also known as descriptive models. Regularly, unsupervised learning is used to strengthen the researchers' understanding of the data before actually using it to build predictive models. For example, researchers may cluster data, as exemplified by Figure 3, to see if patients who did react to some drug share some specific similarities versus patient who did not, in order to evaluate which supervised method would be the most appropriate to build a drug-response estimator.

Considering this work is focused on building predictive models, supervised learning is further detailed in the next sections.

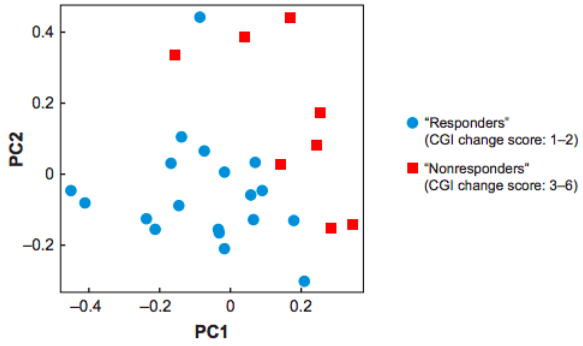

Figure 3: Lipidomic analysis of data for schizophrenic patients treated with atypical antipsychotic drugs. Principle component analysis was then applied to these metabolites, and separation of the groups was visualized with a scatter-plot of the first (PC1) versus the second (PC2) principal component, where the red squares represent subjects who responded to drug treatment and blue circles those who did not respond to drug treatment. This picture is from a study which can be fully read on [15].

2.2.1 Supervised Learning

Machine learning (ML) can be shortly defined as the ability to generalize a task, from and beyond examples: examples are modeled, and the resulting model is expected to successfully foresee the task in question when seeing new example data.

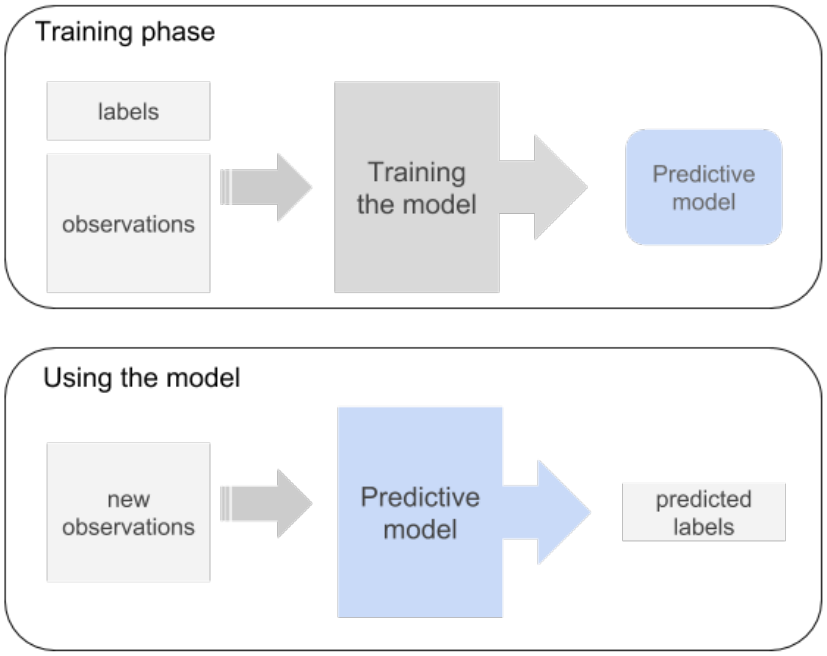

Figure 4: Flowchart illustrating the training of a predictive model (top) and the using of a predictive model (bottom).

In supervised learning (SL), the data being used to build models has been

previously labeled, i.e, each example is an observation (input) associated with a label (output), as shown in Figure 4, and the goal is to optimize an input-output mapping function, which is commonly known as a model. Optimizing such function is regularly referred to as training the model and it can be simply explained as the determination of some mathematical function's parameters; different mathematical functions mean, therefore, different SL methods.

After the training phase, the predictive model is then ready to be used: given a new observation, it will output the corresponding label, as it is also shown in Figure 4. When the labels are abstract tags, like cancer or no cancer, we are in presence of a classification problem; if, on the other hand, the labels are real numbers, like blood glucose (sugar) values, which normal range goes anywhere from 3.9 to 5.5 mmol/L , we are in presence of a regression problem.

Although the general idea of SL is quite simple to understand, there are a few issues that invariably require scrutiny when actually building a model.

The supervised methods have remarkably evolved and are capable of modeling complex relationships within the data. Such ability may impress, but unfortunately, oftentimes it means that algorithms elaborate complex patterns that are unique to the data being modeled, and not at all reproducible, which may lead to models with little predictive power.

The whole point of building any predictive model is to encapsulate some knowledge that is believed to be general enough, within a certain context. For example, if we build a model for classifying a tumor as benign or cancerous, it is expected to fairly do it so, for any future tumor data it might be used on.

Overfitting is precisely the inability of a model to maintain the performance levels achieved in the training phase, when it is fed new data, due to its resulting model over detailing patterns found in the training data. Thus, such model is likely to poorly perform on unseen data, even though it appears to successfully classify training data examples. On the other hand, too simplistic models may not find clear trends in the data, leading to models that underfit.

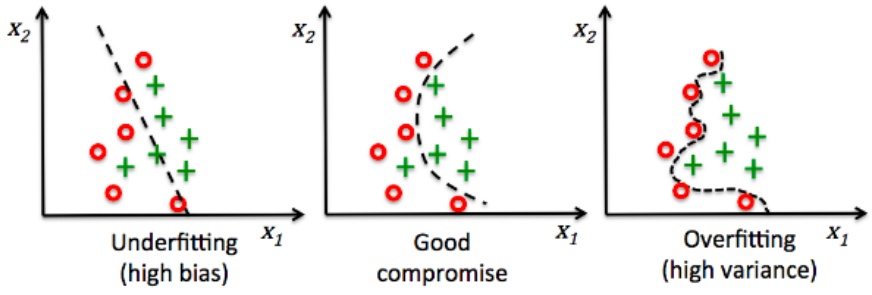

Figure 5: Bias-variance trade off: illustration of high bias (left), high variance (right) and a compromise between them (middle). Picture from [16].

Both these antagonistic characteristics disrupt a model's ability to accurately perform, because a model that cannot find evident patterns in the data is just as useless as one that considers noise as part of the patterns themselves, as it is shown

in Figure 5. Finding the balance between both extremes is well known in the ML field as the bias-variance trade-off.

As mentioned, erroneous predictions can be originated by the model's inability of finding existent patterns in the data or the ineptitude of adapting to slight variations around the patterns it learned during training. On other words, the performance of a model can be decomposed on bias and variance errors: bias refers to the assumptions that the model encapsulates, whilst variance refers to the sensitivity of the model[17].

Considering that bias relates to the ability of the model to approximate the real data, high bias leads to underfitting, and is the case where the model failed to learn clear patterns in the data. For example, modeling a quadratic relationship with a linear regression will always lead to high bias. On the other hand, variance indicates the stability of the trained model before new data. Thus, high variance expresses the model's inability to generalize the patterns found, and logically tends to increase with the model's complexity.

As it should be clear by now, ideally, any predictive model would exhibit low bias and low variance. However, as that is rarely the case, researchers seek the best compromise possible, as the center picture in Figure 5 demonstrates.

Given the issues discussed above, it is a standardized practice to test different models in the data, meaning different parameter values, and chose the one which displays the most robust performance. A common strategy for model selection is Cross-Validation (CV) and consists on splitting the data into separate sets, using different data for training and testing the model[18]: part of the data is used to train the model, called training set, and the rest of the data is used to assess the predictive performance of the model on unseen data, known as the test set. Naturally, the model that displays the better predictive power in the test set is usually the selected one.

2.3 ML methods

There are numerous classifier algorithms that can be used for building predictive models [19]. Detailing different algorithms is not the purpose of this project and, thus the next sections exclusively, and briefly, present the supervised learning algorithms explicitly used in the final prototype of this work.

2.3.1 Support Vector Machines

Originally, support vector machines (SVM) is a binary classifier, which means that each input in the data is, or will be, associated to one out of two possible output classes. Specifically, the idea of SVM is to compute a hyperplane capable of separating the samples of both classes, with the largest margin possible [20].

As Figure 6 illustrates, the points closest to the hyperplane are called support vectors, and they basically are the hardest points to classify, as they are the closest to the separation boundary that is the hyperplane. The distance between support vectors and the hyperplane is called margin, and SVM performance highly depends on its ability to maximize this margin.

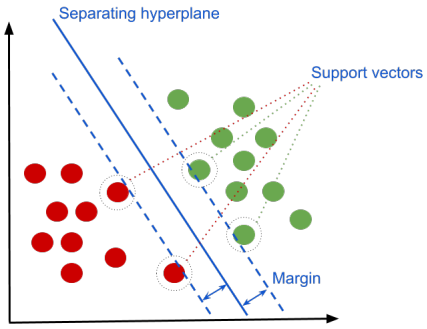

Figure 6: Illustration of SVM elements: hyperplane and support vectors.

In most real life classifying problems, however, classes are not linearly separable, as there is no linear equation that perfectly separates both possible outputs. Therefore, in science, it is customary for researchers to develop new formulations of known algorithms to overcome some specific limitations, and SVM has indeed been adapted for non-linear scenarios.

One technique is to simply relax the original algorithm's objective function [21], and shifting the goal to find a reasonable enough separating hyperplane, by balancing the maximization of the margin whilst minimizing the quantity of misclassifications [22]. This is achieved through a regulating parameter, C, which value is set by the user, and it essentially represents the trade off between allowing training errors and forcing rigid margins. Low values of C allow more training errors while high values allow the model to select more support vectors, in order to guarantee the correct classification of the entire training set.

Equation 1 formally details the objective function of SVM, with the mentioned regulating parameter.

$$\min \left(\|w\|^2 + C \sum \xi_i \right) \tag{1}$$

, where w represent the weights that are actually learned during the training phase, ξ the loss function value, and C the regulating parameter.

For further details on SVM and its parameters, we recommend the read of [23].

2.3.2 Kernels

Still regarding the non-linear separability of real-life data, the most common choice is not to simply relax SVM original constraints. An almost standardized strategy is to map the data into a higher dimensional space, where the points of each class are indeed separable, as Figure 7 exemplifies.

A kernel is precisely a function that maps the data to a feature space. Following [25] notation, a kernel is defined as a function k, that for all $x, z \in X$ satisfies:

$$k(x, z) = \langle \phi(x), \phi(z) \rangle \tag{2}$$

, where ϕ is the actual mapping from X to an inner product feature space F.

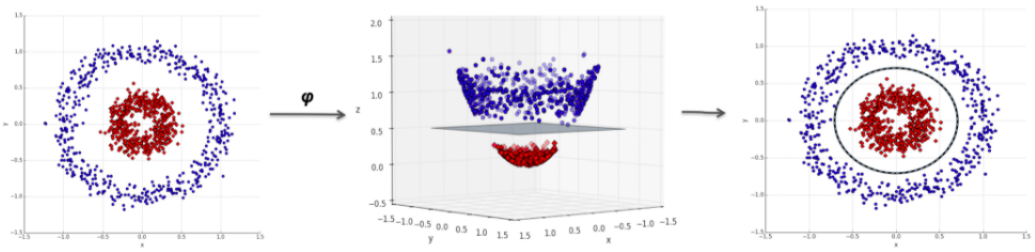

Figure 7: The kernel trick maps the data into a feature space where the two classes are linearly separable, through a non-linear mapping function ϕ. Example based on [24].

One major advantage of using kernel functions is the circumvention of expensive calculations in high dimensional features spaces, by avoiding the need for actually computing the new feature space, which the classifiers would then use for training, by implicitly representing them strictly by those space's inner products.

There are a few kernels that are regularly used in SL problems, from which we highlight the linear kernel, polynomial kernel and Gaussian kernel. There is no universal procedure to follow when choosing a kernel function, as it greatly depends on the data structure [26], hence researchers usually end up empirically selecting which kernel to use, for a specific problem.

As the Gaussian Kernel (or RBF) was used in this work, a short description is now presented. Formally, the RBF kernel is defined as

$$k(x, z) = \exp(-\gamma \left\| x - x' \right\|^2) \tag{3}$$

, where γ is the parameter the sets the range of the kernel and it is usually not manually set, but instead selected by cross-validation. Larger values of γ will lead to more narrow Gaussian distribution, and smaller values with naturally lead to wider distribution, as it is shown in Figure 8.

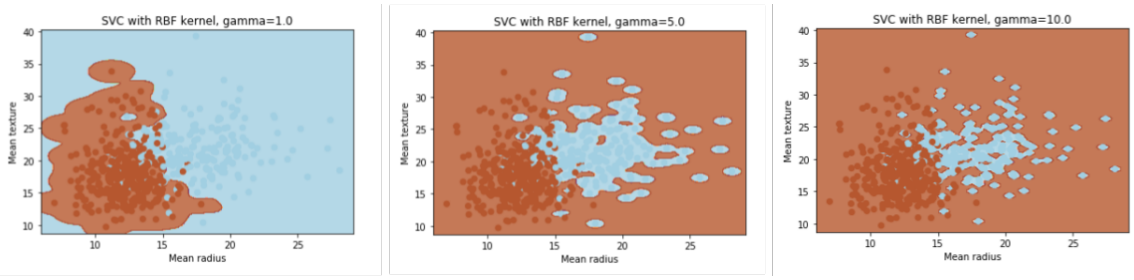

Figure 8: Demonstration of the effect of the parameter gamma, γ, in the RBF kernel, using SVM algorithm. Data from the breast cancer dataset avaible on Scikit-learn[27].

The review of all kernel functions is out of the scope of this work, and for a deeper understanding of kernels, we recommend the full read of [25].

2.3.3 Decision Trees

Decision tree (DT) is a notorious supervised learning algorithm that arranges data in a comprehensive tree-format, where branches are basically paths that lead to certain outcomes, represented in the tree extremities, the leafs.

As Figure 9 illustrates, the simplistic representation of DT closely mirrors the human classification process, where patterns are identified through a sequence of questions, in which the previous question influences the next one. Such scheme makes binary DT easy to interpret, even by non-experts, which cannot be said about many predictive algorithms.

The process of building a decision tree essentially consists on organizing feature variables on a certain order [28]. The algorithm recursively splits the training data into smaller subsets, according to certain heuristics and, naturally, finding these heuristics is one of DT's biggest challenges.

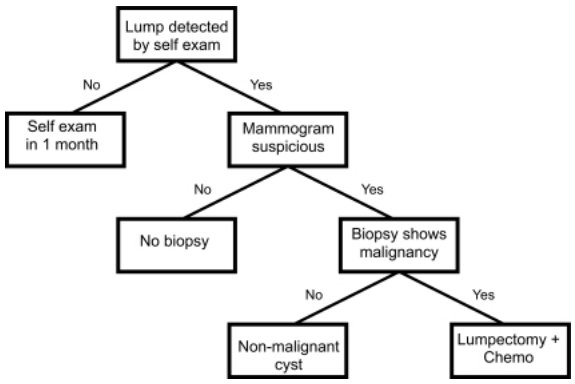

Figure 9: An example of a simple decision tree that might be used in breast cancer diagnosis and treatment [29].

Ideally, each heuristic splits the data into homogeneous subsets, which would ultimately lead to the leaf sets to have a unique outcome class. Unfortunately, ideal is oftentimes not the case and thus, in order to construct a satisfying DT, the homogeneity of sets is tracked throughout the building process, and it's referred to as impurity. Instinctively, the goal is then to minimize the weighted average of the impurity of the resulting children nodes [30]. Different impurity measures have been designed, out of which the Gini index and entropy are the most commonly used by DT algorithms. The detailing of impurity measures is out of the scope of this thesis, and for a better understanding of the matter, the full read of [31] is recommended.

An important aspect to consider when building a DT is its final complexity, as too deep and therefore complicated trees, will have easily overfitted the training data. The resulting DT are supposed to be general models of patterns and not simply visualization tools for training data. There are a few approaches to avoid this issue, namely stopping the splitting of data when no heuristic increases the purity of the subsets, or to prune some branches by deleting some lower levels of the tree, making it more generic [30]. For illustrating purposes, Figure 10 shows two examples of decision trees that were manually pruned, i.e. trees which the last level(s) of nodes

was purposely discarded. However, the pruning is often given by the complexity parameter which is fed to the decision tree algorithm, which only outputs the already pruned tree.

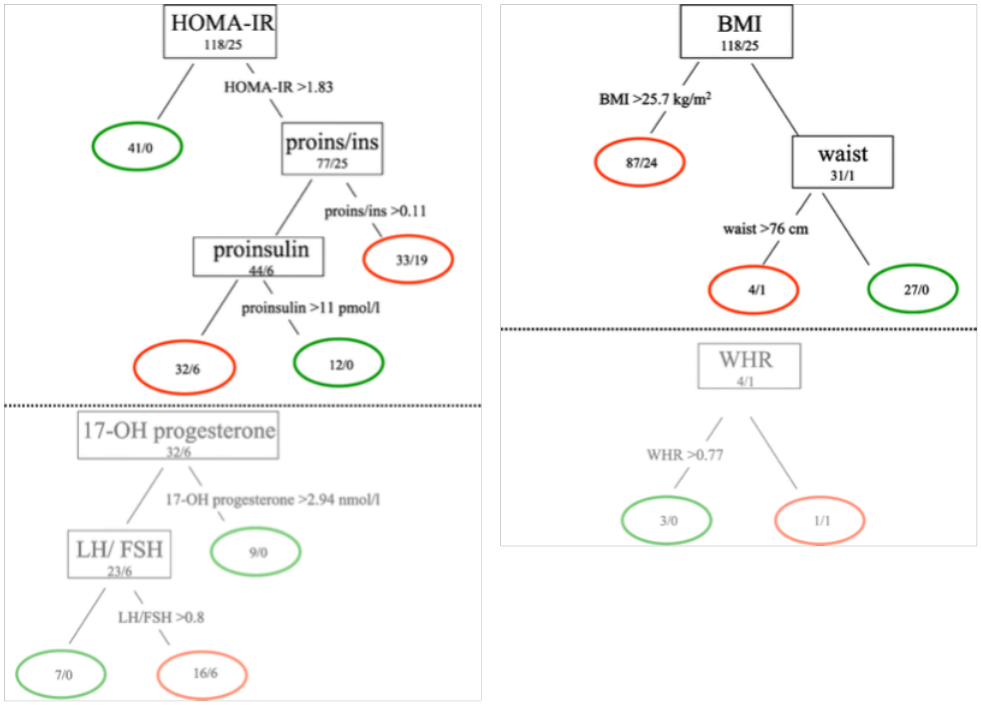

Figure 10: Example of decision trees that were manually pruned, with the horizontal line indicating at what level. The x/y values inside the nodes indicate the total number of cases with the condition of interest and the red color indicates the presence of those cases within the node. For more, the full study can be read at [32].

In addition to the already mentioned simplicity and transparency of resulting models, the increasing popularity of DT in research can be further explained by its modeling versatility, dealing with numerical, categorical and/or mixed data, besides handling data with missing values.

DT have consistently performed beyond expectations, displaying much more predictive power than initially expected out of such a simple method. Moreover, more recent approaches of aggregating several DT, such as boosting or random forests, have yet again over-performed many contemporary and complex predictive algorithms, and are nowadays considered to be among the SL algorithms with best predictive performance [33].

3 Related work

3.1 ML in health care

Health care encompasses many different fields of research, but when talking about ML, Medical imaging (MI) is clearly one of those fields that has benefited from ML research, by significantly enhancing already-in-use methods, specially in the area of diagnosis imaging (DI), and brain function mapping [34].

MI, in general, and DI, in particular, naturally depend on the quality of the collected images, which has undoubtedly evolved as well, which of course contributed for the success of different ML methods that were developed and used [35]. This is of particular relevance considering that the early - and most modern - ML methods applied in the DI field follow the supervised learning paradigm, which is heavily dependent on data quantity and quality, as it was described in the previous chapter.

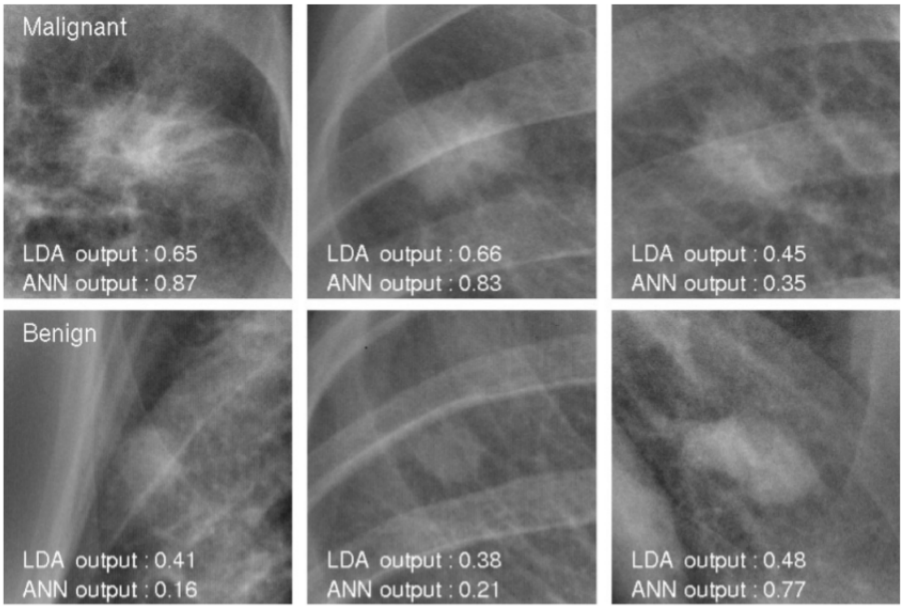

Figure 11: Original description: Illustration of malignant and benign nodules on chest radiographs together with the likelihood measure of malignancy obtained with a computer-aided diagnosis (CAD) scheme by use of linear discriminant analysis (LDA) and on artificial neural net- work (ANN). A computer output above or below 0.50 indicates the likelihood of malignancy or benignancy, respectively. Picture from [36].

If initially the believe was that computers would end up completely replacing doctors, the practice has evolved into the the heavy use of computer-aided diagnosis (CAD), which broadly consists on using a computer output as a second opinion of a physician's interpretation of images. Computationally, a CAD system searches for lesion regions and assesses the likelihood of a disease based on them, for which different tasks are required, such as image processing for detecting the anomalies and data processing for classifying those anomalies [36]. Examples of diagnosis research that incredibly evolved with those are breast, pulmonary and colon cancer researches, nodules detection [37].

The ML methods commonly used in CAD were initially mostly linear, such as linear discriminant analysis, naive Bayes, K-nearest neighbors, with the non-linear approach of support vector machines and neural networks making a significant impact in the field [38] .

More recently, deep learning has been heavily researched for the improvement of many state-of-the-art MI systems, which is not a surprise, considering the impressive results this approach has achieved throughout different fields, such as image recognition, natural language processing, speech recognition, self-driving cars, among many others [39].

The deep learning paradigm allows for the automatic optimization of how data is represented and consequently used my the algorithms themselves. Particularly in field of image processing, algorithms like convolutional neural networks (CNN), for example, have been achieving impressive results, whilst simultaneously simplifying the training processes[40]. For a more detailed review of how it has been applied in the MI fields, please read [41–43], as deep learning is outside the scoop of this thesis.

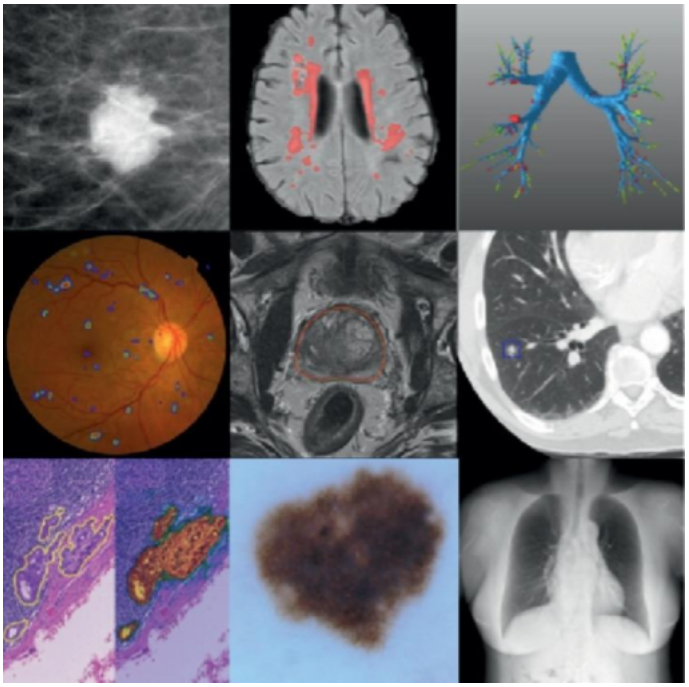

Figure 12: Original description: Some examples of ML achieving state-of-the-art results in medical imaging applications. From top-left to bottom-right: mammography mass classification, segmentation of lesions in the brain, leak detection in airway tree segmentation, diabetic retinopathy classification, prostate segmentation, nodule classification, breast cancer metastases detection, skin lesion classification, bone suppression. Picture from [40].

Regarding brain function mapping (BIM), and most neuroscience research fields, in addition to the image processing challenges, and solution approaches already mentioned, there is the samples versus features dimension issue. For instance, in a BIM scenario, the collection of data is oftentimes done by recording 128 electrodes, for a certain period of time. It is easy to understand that in theses cases, one patient's

input has a huge dimension, whist the study sample can be on the order of hundreds. Thus, working around this sample-feature unbalance is always done, and it may involve spatial, spectral, and temporal pre-processing of the input. Unsupervised learning algorithms are used for these tasks, of which common choices include Principle Component Analysis (PCA), Independent Component Analysis (ICA), Non-Gaussian Component Analysis [44], just to name a few. As unsupervised learning is beyond the scope of this project, it will not be further detailed.

3.2 ML and clinical data

Clinical data refers to ongoing patient care information, such as electronic health records that may include demographics, diagnosis, drug prescriptions, laboratory tests, among other types of information [45]. Besides genomic data, the amount of clinical data being deliberately collected by biomedical research, and heath care facilities, have increased significantly in the last few years. Machine Learning (ML) has greatly contributed for the progress of clinical medicine, as it has been developing methods that aim at improving the extraction of knowledge from all these available data, often leading to performance enhancements of previously existing systems. In fact, ML was initially mainly focused in disease diagnostics, namely of several cancer types [4], and its focus was later broaden to disease prediction and prognostics, which is the scope of this thesis. It is worth saying that predictive medicine may deeply impact, not only physicians' treatment decisions and patients' lifestyle choices, but also health policies, namely regarding diseases awareness and prevention campaigns or even the implementation of specific treatment policies [46].

For the last couple of decades, numerous ML methods have been used in biomedical research for disease detection and diagnosis, as well for prediction and prognosis. Accordingly, feature selection techniques have also vastly evolved, specially regarding genomic data, of which studies [47–49] are an example of.

Regarding disease prediction and prognosis, the most recurrent research targets are disease susceptibility, recurrence and survivability, with cancer often being the condition in the spotlight, representing an impressive slice of predictive biomedical groundwork. Susceptibility is the likelihood of developing a disease, recurrence is the prospect of a disease developing once again after its apparent eradication, and survivability is about predicting life-expectancy, disease progression or disease sensitivity after the diagnosis [29].

Based on the focus of this work, the study of disease prediction, four major ML techniques were identified in the literature [50]: artificial neural networks (ANN), decision trees (DT), support vector machines (SVM) and Bayesian Networks (BN), with ANN and DT undoubtedly being among the first algorithms applied to biomedical research.

The concept of ANN is based on the human brain functionality, and they are frequently used in predictive modeling, as they have been consistently shown to fairly capture highly complex patterns in the data. ANN have been used to predict the occurrences of various diseases, such as cardiovascular diseases, endless types of cancer, among others [51]. On the other hand, DT are a rule-based approach that

progressively splits the data according to identified trends, consequently generating models that are easy to understand. An overview of medical applications using DT can be consulted in [52].

More recently, SVM and BN have been applied to disease predictive modeling, having often achieved impressive results. In short, SVM takes high-dimensional data and attempts to find a hyperplane that separates the different classes, with as large a margin as possible. SVM have been successfully used with both genomic [53] and clinical data [54], having lead it to be one the favorite algorithms to consider when building predictive models, in medical research as well as on other contexts.

A BN is a probabilistic graphical model that expresses variables and their dependencies, via a directed acyclic graph. They have become increasingly popular mainly due to their demonstrated ability to handle the uncertain knowledge involved in establishing diagnoses of a disease [55]. Several real-world applications of BN in bio-medicine are reviewed in [56].

3.3 Longitudinal cohort studies

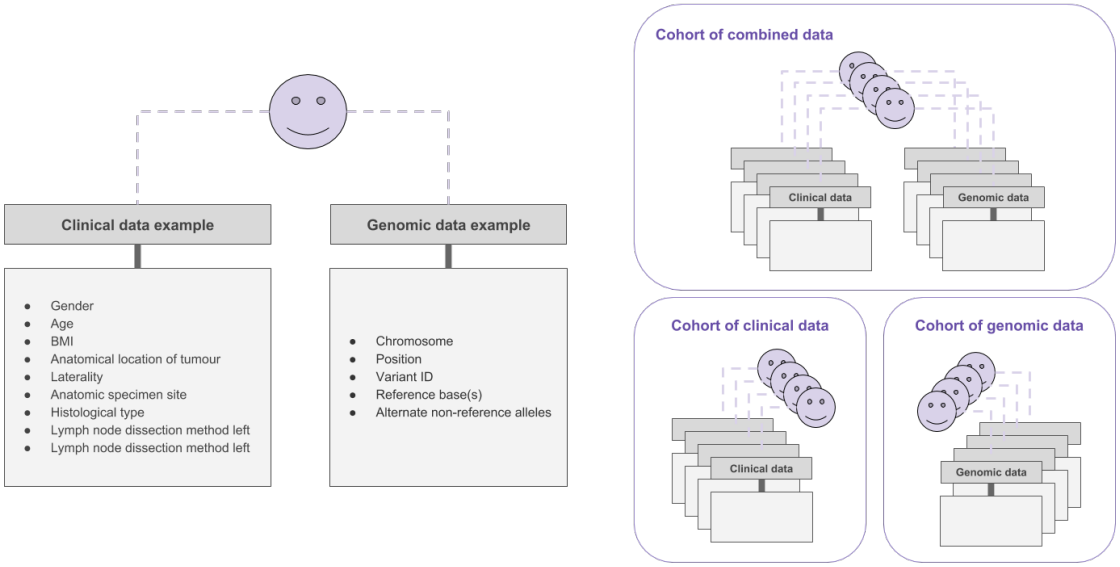

Figure 13: Illustration of the concept of cohort data. Left picture: description of one sample, the unit of cohort data. Example of both clinical and genomic data. Right picture: illustration of cohort data.

The choice of method(s) to use in a certain study is naturally influenced by the structure of the data. Cohort data refers to samples data and, in the context of this work, it may refers to clinical, genomic or combined data, as Figure 13 illustrates. The majority of cohort studies use two-dimensional data, meaning that each patient is represented by one sample. However, high-dimensional cohort data has been progressively more used, with each patient being represented by several observations, usually repeated measurements of the same variables, over a certain time period.

If the samples are collected over a long period of time, we are in presence of a longitudinal cohort study.

Cohort data, given its time-series nature, is likely to contain subtle patterns that may be quite informative in respect to patients physiological state and, therefore, their long-term outcome. There are two main types of analysis implemented on cohort studies: survival analysis or the already mentioned predictive analysis.

Survival analysis regards a collection of statistical procedures for analysing data where the outcome variable is the time until the occurrence of a certain event [57], that could be either death, divorce, market crash, or any other. In the clinical context, survival analysis is of great interest, as it gives important insights into the evolution or regression of patients' state throughout time, for example, if treatment is indeed being effective or not [58].

Straightforward predictive modeling is, nonetheless, still extensively used with high-dimensional cohort data, either by analysing data in its time sequence format, or by transforming it into two-dimensional data through either feature selection or feature extraction techniques. Time-series classification methods have, as a result, been increasingly integrated into longitudinal cohort studies and their approach is inherently related to similarity metrics. There have been studies published where this methods have proven surprisingly effective, such as [59]. Bolder approaches have found ways to have standard ML algorithms, DTs for instance, dealing with specific constrains, such as time series data [60].

4 Clinical Data

Healthcare facilities regularly save different types of patient information, from personal demographic details, like age and gender, to specifics about patients' past and ongoing medical care, such as medical history, diagnosis or treatments. Nowadays, all this data is naturally digital and is commonly referred to as electronic health records (EHR) [45]. An EHR includes different types of data, that are frequently being collected from different sources and used for diverse purposes. For instance, personal details may be used mainly by administrative staff, while laboratory results are presumably primarily used by healthcare professionals, like physicians, nurses or pharmacologists.

In biomedical science, clinical trials have been the standard approach for collecting medical data, on which studies are subsequently based upon. However, this approach has always had the inconvenience of being too time consuming, expensive and condition specific, as a result of following a group of subjects throughout a certain period of time (that could go from a couple of weeks to a couple of years), oftentimes for a very specific research question [61, 62]. Plus, several ethical questions have been raised regarding clinical trials [63]. All these reasons have made EHR very popular in biomedical research, even though it only allows for retrospective clinical studies, which is a paradigm shift from the classic prospective clinical studies. Healthcare facilities collect EHR anyway as part of their management systems, making them a cheap source of extremely diverse data, especially considering its volume, both patients and measurements wise.

When using EHR as the source of clinical data, there is the additional challenge of the data sources' heterogeneity [64], with different measurements being collected separately and not at all integrated in the database. For example, blood laboratory tests and monitoring devices of vital signs are rarely annotated as possibly medically associated, data-wise. Moreover, there is also the heterogeneity of the type of measurements that could be either numerical, categorical, free-text, or some other type. Researchers must also be attentive to the possibility of erroneous data, missing data and be particularly cautious when using imprecise data, such as data with unknown labels or ambiguous manual annotations. For a more detailed analysis of the challenges of collecting and working with clinical data, we highly recommend the full read of [65].

4.1 MIMIC-III Database

This project was developed using the freely available Medical Information Mart of Intensive Care (MIMIC) database as a source of clinical data, specifically the latest release - MIMIC-III, which is a large, single-center database that includes diverse information regarding patients admitted to critical care in Beth Israel Deaconess Medical Center, in Boston (USA), between 2001 and 2012 [66].

It is worth mentioning that MIMIC is, of course, fully documented [67], and it has been extensively used on biomedical research, as a credible source of clinical data, for studying diverse matters such as the effect of age and other clinical circumstances

on the outcome of red blood cell transfusion in critically ill patients [68] or the trends in severity of illness on ICU admission and mortality among the elderly [69], just to name a few examples.

First and foremost, working with MIMIC solved the complication of integrating different sources of medical data into one single system, which would have been a difficult first step of data pre-processing, with the aggravating potential of influencing the study's results, possibly even invalidating them. Secondly, the actual variety of data per patient, which is summarized in Figure 14, from time-stamped physiological measurements, to physicians' free-text observations, among many others, is of immense value, providing the researchers with an unparalleled multi-dimensional perspective that was not frequently possible with data collected from clinical trials, where usually only few specific measurements that researchers believed to be of interest would be considered. Beyond the wide variety of information, the volume of data available also provides the possibility of using novel data-driven methods of machine learning on a field that has traditionally had limited amounts of data to work with, which can lead to new findings or even deepen our understanding of certain medical conditions, and therefore minimize their mortality rate.

Design Type(s)	data integration objective
Measurement Type(s)	Demographics • clinical measurement • intervention • Billing • Medical History Dictionary • Pharmacotherapy • clinical laboratory test • medical data
Technology Type(s)	Electronic Medical Record • Medical Record • Electronic Billing System • Medical Coding Process Document • Free Text Format
Factor Type(s)	
Sample Characteristic(s)	Homo sapiens

Figure 14: Types of data available in MIMIC database[66].

As it is based on EHR, MIMIC essentially consists on raw medical data, which is of particular interest, considering that many cohort databases available for research have been through unclear stages of data preprocessing. However, the fact that the data is from intensive care units raises some concerns, mainly because these are the most critically ill patients, and being in such health condition may make the data noisier than expected, or not at all appropriate for extrapolating conclusions to non-critical patients.

4.2 Data Structure

Due to the limited duration of this project, not all available data was used in our system. This project solely used numerical laboratory measurements, along with the associated diagnosis information and patients' demographic details, of which a sample is shown in Figure 15.

The table referring to patient demographics, the "patient's table", consists of five variables: the subject unique identifier number, gender, date of birth, date of

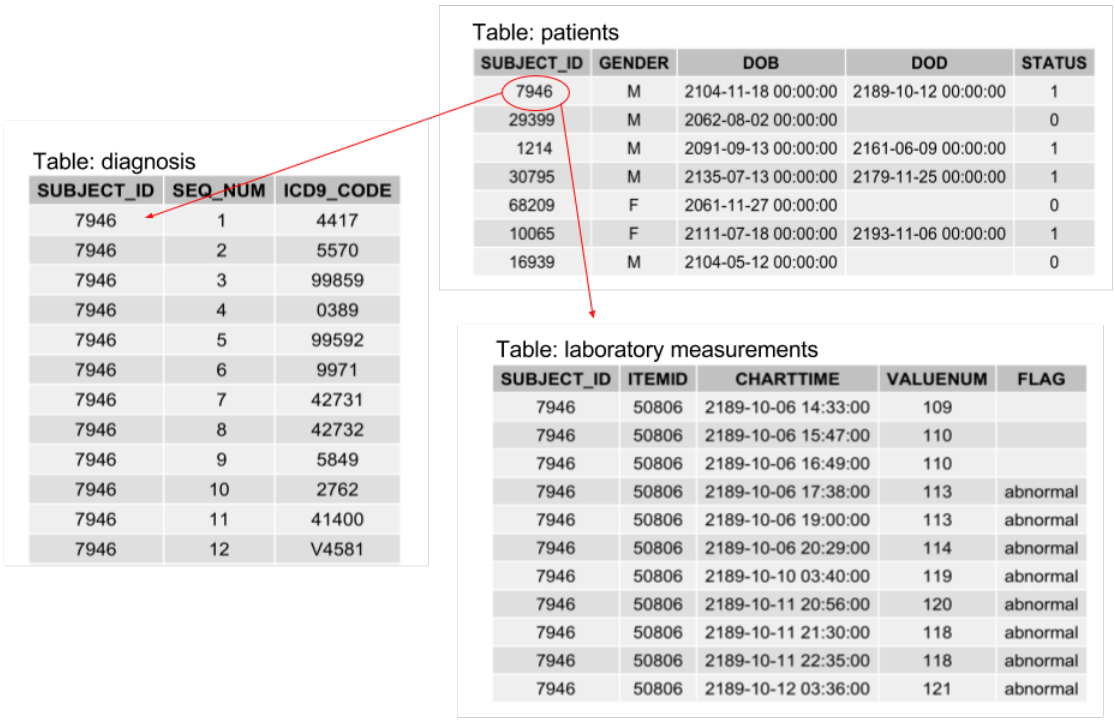

Figure 15: MIMIC's data tables and how they are connected.

death and a status flag that indicates whether or not the patient has died, both within or outside the hospital. The diagnosis table lists all diagnosis the patient was associated with, for billing purposes at the end of the hospital stay, and they follow the ICD-9 (International classification of diseases) codes [70]. Lastly, the laboratory measurement table is the main source of clinical content. It has around 25 million entries, and consists of all measurements conducted, for each patient, of any item, along with the corresponding timestamp. It is worth noting that all MIMIC information regarding dates was shifted years into the future values, fulfilling the required data privacy regulations.

Besides the above data tables, we used the diagnosis and laboratory items' dictionary tables, shown in Figure 16, exclusively for consulting the human-readable labels corresponding to the numeric codes.

The reason for selecting a subset of all available data in MIMIC was two-fold. First, the scope of this thesis automatically excluded data in formats like free-text or imaging data, due to the fact that such approach would require natural language and image processing techniques to convert the data to the appropriate input format envisaged for our system. Secondly, once the data was circumscribed to numerical and categorical measurement values, the decision to drop the latter was mainly due to the fact that numerical data is more versatile to work with, considering the experimental environment where existent methods were used to explore and experiment on data. Therefore, having more diversity of algorithms to understand the data was an important advantage in the early stages of development.

Table: diagnosis dictionary

ICD9_CODE	SHORT_TITLE	LONG_TITLE
0389	Septicemia NOS	Unspecified septicemia
2762	Acidosis	Acidosis
41400	Cor ath unsp vsl ntv/gft	Coronary atherosclerosis of unspecified type of vessel, native or graft
42731	Atrial fibrillation	Atrial fibrillation
42732	Atrial flutter	Atrial flutter
4412	Thoracic aortic aneurysm	Thoracic aneurysm without mention of rupture
4417	Thracabd anurysm wo rupt	Thoracoabdominal aneurysm, without mention of rupture
5570	Ac vasc insuff intestine	Acute vascular insufficiency of intestine
5849	Acute kidney failure NOS	Acute kidney failure, unspecified
99592	Severe sepsis	Severe sepsis
9971	Surg compl-heart	Cardiac complications, not elsewhere classified
99859	Other postop infection	Other postoperative infection
V4581	Aortocoronary bypass	Aortocoronary bypass status

Table: laboratory items' dictionary

ITEMID	LABEL	FLUID	CATEGORY	LOINC_CODE
50806	CHLORIDE, WHOLE BLOOD	BLOOD	BLOOD GAS	2069-3
50807	COMMENTS	BLOOD	BLOOD GAS	
50808	FREE CALCIUM	BLOOD	BLOOD GAS	1994-3
50809	GLUCOSE	BLOOD	BLOOD GAS	2339-0
50810	HEMATOCRIT, CALCULATED	BLOOD	BLOOD GAS	20570-8

Figure 16: MIMIC's dictionary tables.

4.3 Clinical Laboratory Measurements

As mentioned before, this work deliberately considers the laboratory measurements of patients the only source of data. This subsection describes such MIMIC subset, as a context for the system development, which is described in the next chapters.

The considered laboratory dataset reports the measurements of over forty-six thousand patients, from neonatal to elderly patients. However, considering the research goal to automatically mine the clinical dataset, it was a conscious decision to consider only patients from the age of ten onward, as well as excluding patients over ninety years old. Such compromise was achieved considering that both these groups of patients are potentially admitted into ICU given some very specific medical conditions. For instance, premature newborns have an even higher risk of survival given their physiological development might not be complete, or very old people's recovery might be less likely given their advanced age. Thus, only patients from 10 to 89 years old were considered, which we catalogue as adult patients, and Figure**??** shows this dataset age distribution, as well as the correspondent survival quota.

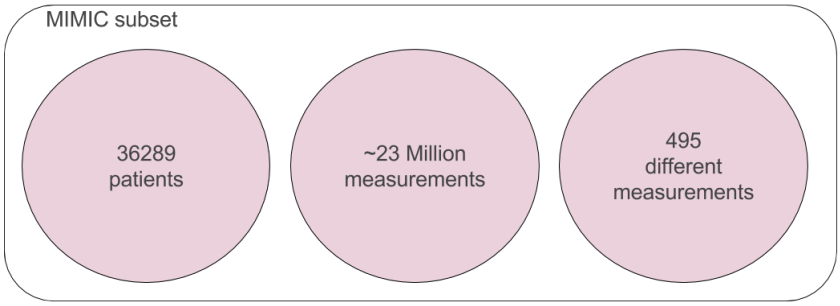

Figure 17: Generic metrics of MIMIC subset considered in this work, before any pre-processing.

Figure 17 summarizes the subset of data considered the system's database, i.e., MIMIC data after excluding neonatal and advance elderly patients' related information. From this point on, such subset will be simply refer to as data, as it all the data available to the system that was developed.

Figure 18 illustrates the gender distribution of the data, as well as the survival

quota, showing that the survival quota tend to decrease with age, which makes theoretical sense, considering older people are naturally more likely to succumb to a serious condition than younger people, even considering that intensive care units always involve serious life threatening conditions.

Figure 18: Left picture: Age distribution of adult patients at time of ICU admission[67]. Right picture: Gender distribution and correspondent survival quota of the adult patients admitted into the ICU.

4.4 Data Pre-processing

MIMIC is periodically maintained and updated, which means that most common "contamination" issues of cohort data sets have been solved. For instance, there are very few missing values or imprecise entries. Nevertheless, is usually recommended to inspect the data for such issues anyway, to avoid later set backs. Thus, the pre-processing phase was considerably minimalist.

On the subject of data quality, three simple condition scenarios were used to identify which data entries were invalid and hence to be excluded from further analysis. Each scenario has targeted a different use-case of invalidation and here are their brief descriptions: (1) missing data, specifically measurement data with no item code or no value; (2) ambiguous data, meaning multiple measurement entries for the same patient, same item, at the same time, with different values; (3) imprecise data, as in unlabeled data entries, such as diagnosis with no valid code.

Around sixteen thousand entries were identified and excluded from further analysis, most of which were duplicate entries, making the final processed data numbers shown in Figure 17 unchanged. Even though it may look like a lot of entries excluded, considering the abundance of data available, not much time was spent investigating these dropped entries.

In other respects, the binary variable "FLAG" was initially only filled with one possible value, which makes sense in terms of data storage, but for development purposes, mainly for visualization, this variable was populated with the missing attribute, as it is shown bellow in Figure 19.

The final step of pre-processing concerned the temporal nature of measurement data. As it was mentioned before, and it is shown in figures above, each measurement entry has a time-stamp, represented by the "CHARTTIME" variable, which indicates the date and time of the measurement. However, such detail overshadows the

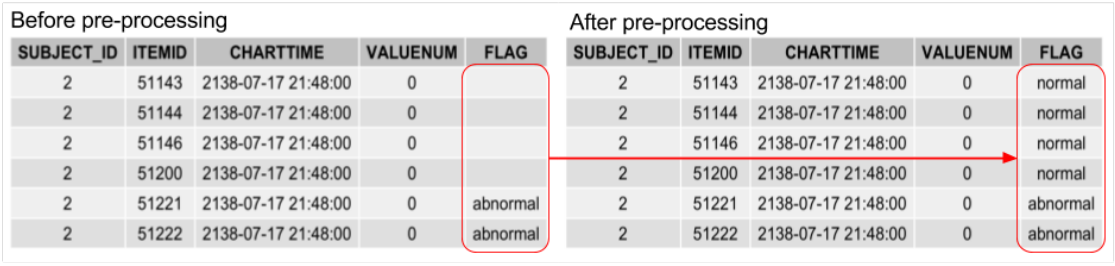

Figure 19: Pre-processing of binary variable "FLAG".

relevant information when the database has entries collected over 10 years. To deal with this phenomenon, all laboratory entries were serialized within the context of a patient, meaning that entries would now have a serial number representative of the i^{th} time that item was measured on that patient. For a better understanding of the serialization, Figure 20 illustrates an example.

Considering this project aims at developing a non-condition-specific data mining tool, this serialization provided the ideal abstraction of the standard timestamps. In this new format, all patient's measurement data are now mapped into an universal chronological scale, which provided an orderly structure for studying the progression of any variable, which is tremendously useful when applying machine learning techniques.

Another aspect of serialization that is worth mentioning is the fact that it does not take into consideration different ICU stays. Thus, after serialization, each patient has an unique sequence of measurements, different for every laboratory item. The belief is that sequential ICU stays may be related, as the human body does not restart after getting better (or even cured), and so, considering all measurements throughout time was an approach that allowed the system to abstract from ICU stays bureaucratic details, such dates and duration, making it focus and use or ignore the measurement values, which is exactly what we want to mine.

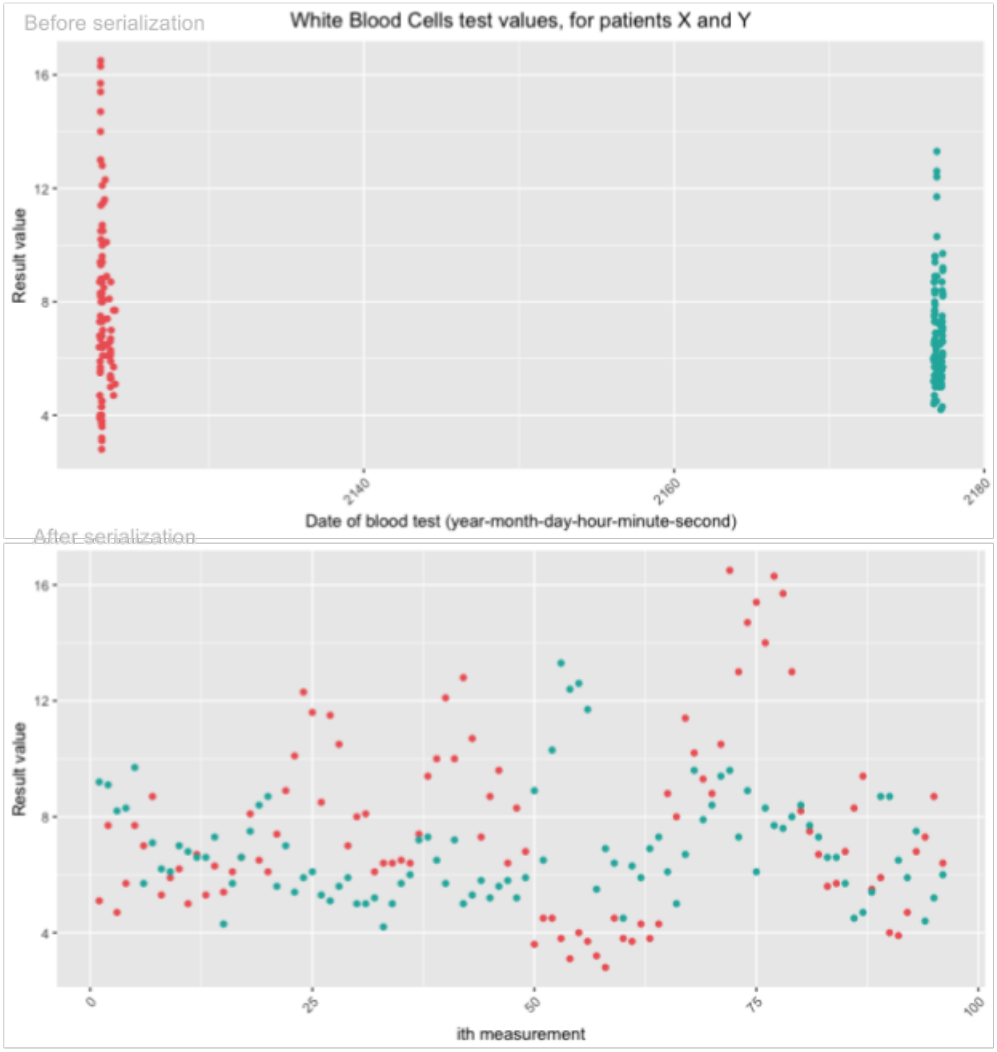

Figure 20: An example illustrating the serialization of measurements timestamps.

5 HypE System

As stated before, the aim of this project is to design and implement an illness-independent system that mines biomedical datasets for potentially interesting medical phenomena. From this point on, for simplicity, such findings will be referred to as hypotheses. Thus, our system can be described as an hypothesis extractor engine (HypE).

HypE was envisaged to be a global data mining tool capable of examine any medical data, but as a proof of concept, it was implemented to firstly require some selection of data, which we call data of interest, and it simply consists on the data relative of a certain scenario to examine. As such, Hype can be seen as encompassing four distinct phases: selection of data of interest, feature extraction, predictive modeling and hypothesis embodiment. Each phase is henceforth detailed. After selecting the data of interest, the first analysis phase concerns feature extraction, which is responsible for transforming the data into a higher-dimensional format, where each entry is represented by several new variables, called features. Next, the newly computed feature data is used to build a predictive model and if the resulting model shows a reasonable predictive performance, it is assumed as indicative of the data containing potentially interesting patterns, in which case the feature data is then translated into an user-friendly hypothesis format.

It is worth clarifying that the contribution of this work is in the pipeline itself, meaning that for each phase, different methods than those used in the implemented prototype can be used, without compromising HypE's premises or purposes.

Figure 21 summarizes the entire pipeline, and the following sections specify all phases in greater detail, as well as present and justify the methodological choices used in the prototype used as base for this work.

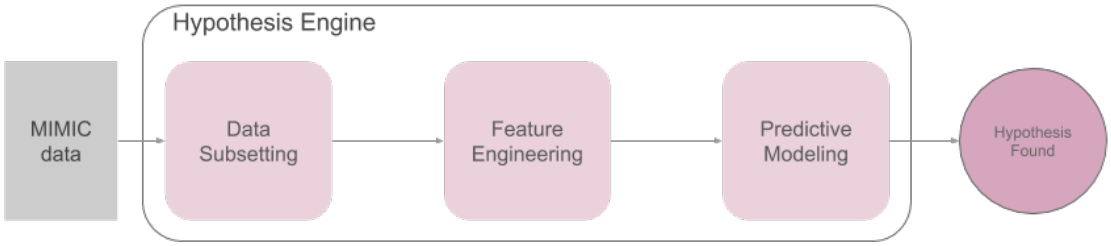

Figure 21: Schema of the HYPE pipeline. Starting with MIMIC database, which is the data source of the system, showing the main three steps of data processing and analysis, up until the output of the hypothesis found.

5.1 Selection of Data

Considering the complex nature of clinical data, detailed in the previous chapter, one way to facilitate its computational analysis, whilst guaranteeing the requirements for valid results, is to restrain each analysis to a consistent scenario. For example, for

investigating if patients diagnosed with diabetes type 2 are more likely to die when admitted into the ICU, HypE would first select all patients with that diagnosis, along with all their medical data, and use only this selection of data in the next phases of the analysis' pipeline. The resulting data of this selection is the data of interest.

The data of interest is, then, a conceptual structure that contains patients with a certain medical condition, for example, patients that share a diabetes type 2 diagnosis, associated with a binary output variable being analyzed that could be, for example, survival outcome of those patients, in addition to all their clinical measurements. As of the time of this document, the system implementation can only assume binary outcome values.

Associating patients to a binary label is a straightforward way to abstractly represent any outcome of interest, from dead or alive, to diagnose A or B, or any other dichotomy. Therefore, this abstraction allows HypE to inspect any pattern of interest, regardless of possible medical specifics of the scenario in the spotlight. Within HypE's context, the data of interest can be simply referred to as data.

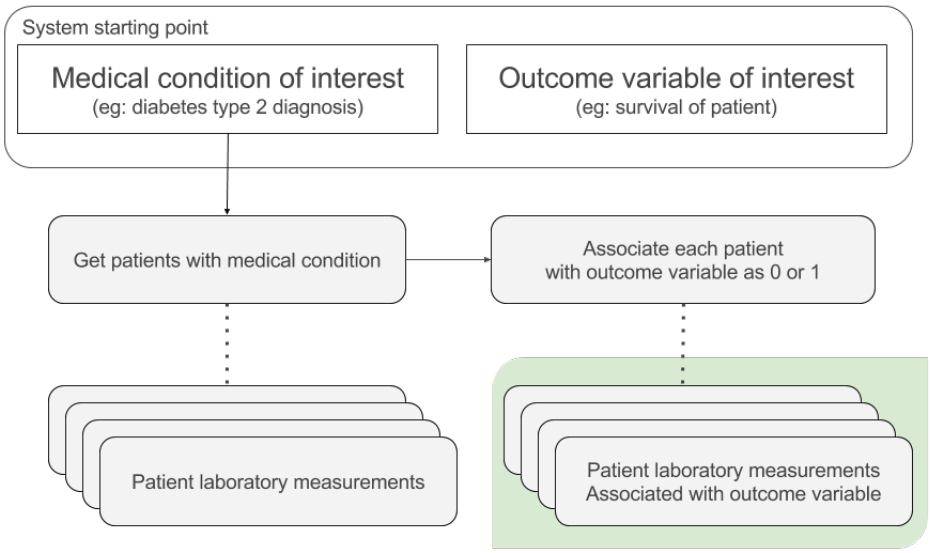

Figure 22: Illustration of the selection of data before getting analyzed. The green box accentuates the data that is actually analyzed for patterns.

Figure 22 illustrates how the data subsets are build out of the all data available. The green box highlights the data subset ready to the analyzed by the system. In case it is not clear, this dataset structure is the same as previously shown in Figure 15, only now, each laboratory measurement is also associated with the outcome being analyzed.

Finally, it is important to mention the requirements that the system has regarding the selection of data. As of now, system has two requirements to accept a subset as valid for analysis, which concern subset size and outcome balance.

Considering the data will be analyzed for its predictive power in the following phase of the pipeline, the size of that data is naturally of importance, as if we have a scenario containing data referring to only 8 patients, the sample size is definitely not

big enough to be tested for predictive power. For this same reason, even in cases where the size of the data subset is enough, it is essential to check if there are enough samples representing both outcomes, as if we have a subset of 300 patients' data, but only 2 of them represent one of the outcomes, any results from modeling the data would be not credible and thus not valid for interpretation.

In the prototype implemented, both values regarding these requirements were previously defined and hard-coded. Therefore, as of now, the data of interest has to include the minimum sample size of 100 patients, and regarding the outcome balance, it was defined that both outcomes must have at least a third of all samples. For instance, if the subset included 100 patients, both outcomes need to have at least 33 samples. If any of these criteria is not met, the system does not proceed with the analysis.

5.2 Feature Engineering

Oftentimes, datasets are a collection of variables, commonly referred to as features, that have clear distinguishable meanings. As it is shown in Figure 15, in the previous chapter, the data has features like patient id, item id, value or timestamp. However, such structure might not be the most appropriate for computationally analyzing the data, which is precisely the purpose of feature engineering.

Feature engineering is a generic term that refers to all methods involved into assembling data with the, as optimal as possible, format for computational analysis, and the resulting data is commonly referred to as feature data. The processing of data may include methods for features extraction, or selection, among other type of restructure. Although feature extraction and selection are often referring to the same operation, in this document, the first refers to the computation of new variables, from the original available ones, while the latter refers to choosing some of those out of all the newly computed variables. For example, a dataset has 10 original variables, out of which 300 new variables are computed, and then, only 52 of those are considered to analysis, thus being the feature data.

It is important to clarify that the feature data is, just as the original laboratory data, organized by variables, but the difference is that in the former case, those variables to not always have to have any human-understandable meaning, as they can be as abstract as the correlation between two variables in a high-dimensional space, for example.

Considering the structure of the data available, as well as the output desired of the system, it was evident that some data processing had to be done. When observing the data, it was noticeable that different items were measured at different rates, and with different periodicity. Medically, measurement rate is influenced by several factors that are closely related to the specifics of a patient medical condition, which is why clinical data is usually so irregular, with many patients outlying out of measurements frequency trends.

Figure 23 shows an example of how the data used for analysis can be visualized, where each line represents one patient. Naturally, given the sequential nature of the data, it is always possible to attempt to manually look for possible patterns that

Figure 23: Visualization of two exemplifying laboratory items as a time-series, for the entire cohort of patients with records for those items.

distinguish outcomes. However, as it shown, it can be messy and unproductive.

The inconsistent nature of clinical data, in addition to the diversity of laboratory items available, complicated the feature engineering, once there was no simple approach to assemble our feature data. Straightforward strategies, such as considering as features all $(item, i^{th} measurement)$ pairs, were simply not feasible considering the result would be a wide and sparse feature set. Having a set with many features is not a problem, on its own, and it is actually an objective of ours, but these features would be mostly empty anyway, with most patients having a different set of items, and frequency, for which they were measured.

Looking ahead, the resulting feature data would be then used to build a predictive model. Hence, our goal was to get a dense feature dataset, while still extracting as many features as possible.

In order to overcome the challenges mentioned above, HypE's feature engineering includes two consecutive methods for extracting and subsequently selecting new features, after restricting the measurements to a certain time range. Regarding the first mentioned segment, which we titled of gradient analysis, we have developed our own algorithm that computes a dense set of features, and then, we use a statistical approach for dropping the newly features that seem to be of no relevance for the posterior analysis.

Figure 24 illustrates the whole feature engineering process, and each step with be detailed in the following subsections.

5.2.1 Time-restriction

The previous chapter explained a time serialization operation that the system performs while pre-processing the data. Although such operation facilitates computational

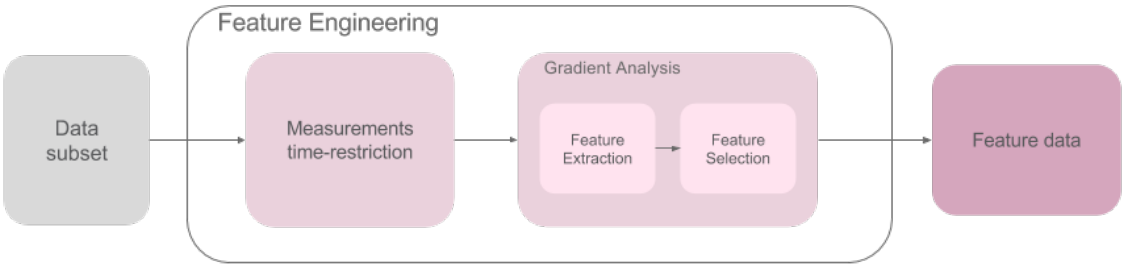

Figure 24: Feature engineering phases: time-restriction approach, followed by the gradient analysis. After, the data is in the appropriate format for analysis, which we designate by feature data.

analysis, there are some cautions that must be taken because of it.

MIMIC's database was collected over a decade, which means that is likely that some patients have been admitted into an ICU more than once, over the course of that decade. This is a problem for two reasons: (1) admissions with years apart make it more likely that different medical reasons were the cause of those admissions; and (2) even if the patient is admitted for the same medical complications, abnormal measurements are likely to lead to medical intervention right away and so, after serialization, it is impossible to know if a certain measurement is constantly showing abnormal values or if some values are just distant in time in the original timestamps.

Few occurrences would be enough to contaminate some outputs of the system and hence, in order to avoid those pitfalls, and considering the prototype implemented solely focus on survival as the outcome variable in question, the strategy is then to restrict the clinical data being analyzed to laboratory measurements showing a timestamp within a month of the outcome. For instance, if a patient is registered as dead, the date-of-death (DOD) is considered to be time point zero, and all measurements taken in the last month are considered; if the patient is not registered as dead, the last day of the ICU stay is considered time point zero. However, after the restriction, the time points are inverted, so that the outcome corresponds to time point 10.

If the outcomes to be analyzed are different that survival status, the time point zero is simply adapted. For instance, if we want to see if it is possible to predict if diabetic patients are likely to be diagnosed with pancreatic cancer, the system can just assume the latter diagnostic date to be time point zero.

In addiction to the month restriction, the system makes use of the last 10 measurements, within that month range. It is set up as so due to the conviction that the closer to the outcome, the more differences must be identifiable within the outcomes. In the prototype implemented, both the month time window and the the amount of measurements considered are hyper-parameters of the system, and thus can be changed by the user at any given time.

5.2.2 Feature Extraction

HypE's prototype was implemented to search for simple patterns, specifically the increase, decrease or stability of the value of any item, that seemed predictive of a certain outcome.

Accordingly, we have developed a method for extracting new features that essentially consists on representing a variation of an item as a single value, by computing the slope of the item variance, between two given time points. This process is repeated for every two time points combination, following a sliding window approach, and for every item, for each patient. In the resulting feature set, each patient is then represented as a group of variables that express how much items values have changed overtime. Figure 25 shows the extraction of feature method.

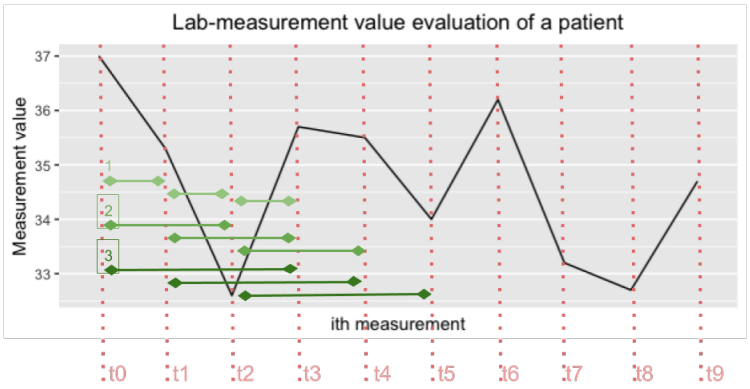

Figure 25: Illustration of the sliding window approach for extracting features: for each two time-points, the gradient is computed, as registered as one feature.

It is worth recalling that in the data of interest, each patient is associated with a binary label, which represents one out of two possible outcomes being analyzed. For example, the label can express dying or not at the ICU. Therefore, having the new variables computed, the feature data has now a two-dimensional format, where each patient is represented as a collection of newly computed variables, and is identified with the corresponding outcome label.

5.2.3 Feature Selection

When data sets have a substantial amount of features, it is frequent that not all of them express relevant information. In order to minimize the chance of certain features polluting the overall appliance of the set, it is common to submit feature sets to some selection method, where they are subjected to a validation analysis, and naturally, only the ones that seem suggestive of being somewhat interesting are kept for the final feature set.

In HypE, the validation analysis of features is a purely statistical mechanism. Each feature is mapped into two cumulative distribution functions (CDF), each one representing a label of interest data, and a feature is considered relevant if the difference between those two distributions is statistically significant.

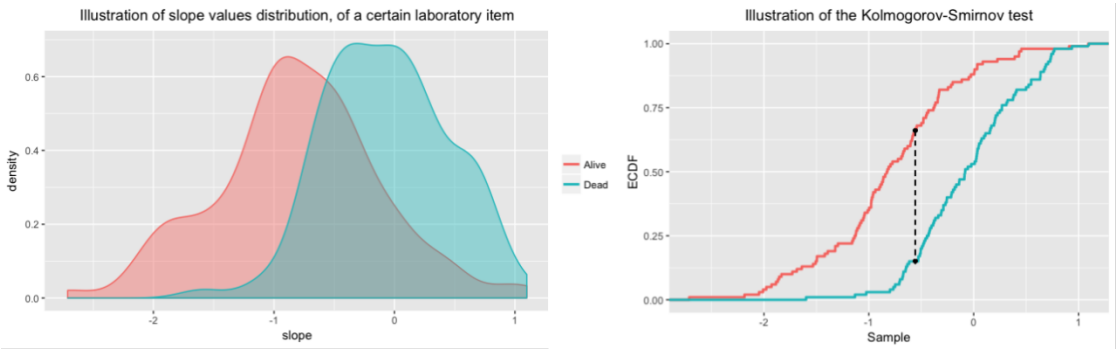

Figure 26: Left picture: the illustration distribution of the data of a certain item, were all patients are considered. Right picture: the representation of the same data as the left picture, but as a cumulative distribution function (CDF). The line on the plot represents the maximum point in which the distributions are different, computed by the Kolmogorov-Smirnov test. Color code: red represents the data of outcome variable "alive" and by opposition, the blue represents the data of outcome "dead".

The assessment of significance is achieved through a two-sample Kolmogorov-Smirnov (KS) test statistic. In short, two-sample KS test is commonly used for comparing two continuous populations [71], in our case, two CDFs. If there are enough discrepancies between the samples, they are assumed to be have drawn from different distributions, in which case the feature being analyzed is considered relevant. The presumption is that if the label-based CDFs are different enough, it is highly suggestive that mining algorithms might be able to identify label-unique patterns within that particular feature, making it an important feature to preserve for the final feature set, as HypE's intent is precisely to find trends, within items, that might be predictive of patients' aftermath.

After the selection of relevant features, each patient is then represented by a smaller amount of features than before, but all of which seem greatly informative, finalizing the processing of the feature data.

5.3 Predictive Modeling

The purpose of this stage is to inspect the data for its predictive power. The conviction is that the model performance is indicative of the presence, or not, of distinct patterns that seem indeed associated to the patient's outcome. Hence, if the model exhibits promising forecasts, the data is flagged as containing interesting phenomena, and is afterwards subjected to the hypothesis extraction process, where these phenomena are encapsulated into an hypothesis. This section details the modeling process, as well as how the assessment of interest is achieved.

First of all, bear in mind that different predictive algorithms have different data format requirements, especially regarding missing values. Some algorithms have not been designed for dealing with be absence of data appropriately, which may obstruct the resulting model in unpredictable or even non-understandable ways. For example, considering missing values as zero is a simple yet brutal mistake that might

make models highly untrustworthy because they will not distinguish those cases from features originally with the actual value zero, misleading the training of the model.

In HypE's prototype, the method selected for the predicting modeling was support vector machines (SVM), which in its original implementation does not support missing values [72]. Thus, in order to guarantee more authentic results, the data to be used for training the model should be complete, i.e., the data should not have missing values. For that reason, the system does include an extra-step for discarding observations with missing values, which is detailed in the next sub-section.

5.3.1 Missing values

After the extraction and selection of features, processes described in prior sections, the resulting data is quite dense, as Figure 27 simplifies, meaning that most patients do have values for most features. However, a completely full dataset is, indeed, rare, which makes sense given the nature of clinical data, where the follow up of measurements is directly related to patients specific clinical status, and not necessarily connected to the outcome label being analyzed within a specific condition of interest.

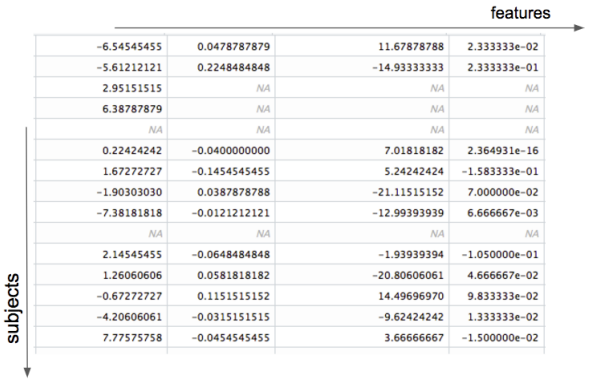

Figure 27: An example of feature data: each line is a subject, represented by several features values.

In the interest of removing the missing values from the feature data, HypE has its own method for sub-selecting a full dataset, out of the original feature data, and that method was developed with the intent of minimizing the amount of data to be discarded. A simple approach to the problem would have been to drop all patients (or features) that displayed missing information, but such solution would often result in small complete datasets, as Figure 28 illustrates, occasionally so small that not enough data was available to properly train the predictive model.

Moreover, when looking at the missing values, it was noticeable that some features usually contain much more missing values than others, which means there are groups of patients that were clearly not measured for some items. For example, a possible explanation might be that such features relate to some laboratory item that was extraordinarily followed up for some patients, for some medical reason. Thus, after exploring the distribution of missing values, for several datasets of interest, it seemed

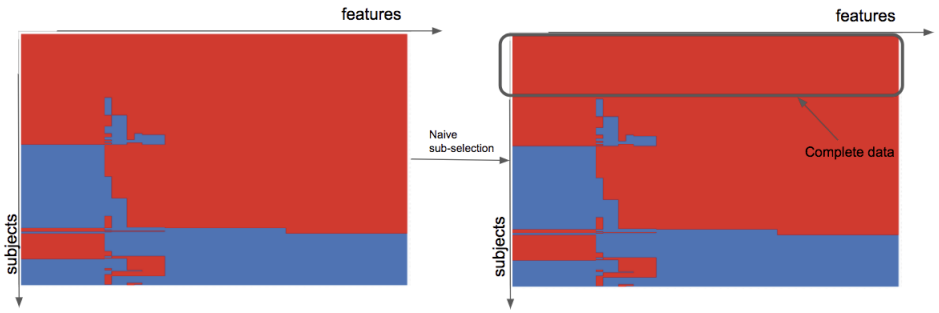

Figure 28: Heapmap of missing values (blue colour), and the corresponding complete sub-selection of data, following the naive approach of excluding all patients with any missing values.

logical to systematically discard features that carry a significant quota of mission values. In the current prototype, all features that have more than 30% of missing values were automatically discarded.

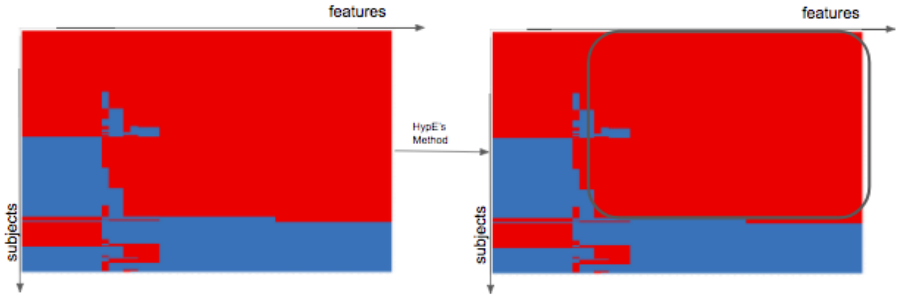

Figure 29: Heapmap of missing values (blue colour), and the corresponding complete sub-selection of data, following the Hype's approach of firstly discard features with a high quota of missing values.

Figure 29 illustrates that starting by excluding features with a considerable amount of missing values, and only then excluding patients with missing values, results in a bigger complete dataset, as intended. For disclosure, it is important to highlight that this cutting-off approach was only feasible due to the abundance of data available on MIMIC, and of course considering HypE was designed to be used on big biomedical databases. If this would have not been the case, in scenarios with limited data available, non discarding methods would have been considered, such as imputation [73], where missing values are replaced by some estimates.

5.3.2 Support Vector Machines

As mentioned before, SVM was the algorithm chosen for modeling the clinical data. Briefly recapping, SVM is an algorithm that finds a boundary, called hyperplane, in the parameter space that better separates the label cases [74], corresponding, in this

project, to examples like "died" or "survived". Computationally, mapping all data points into a higher-dimensional space can be heavy if the data is big enough, which is why SVM is commonly paired up with a kernel function that computes the dot products of data points directly in the high-dimensional space, based on which the hyperplane is then determined. For a more detailed explanation of SVM and kernels, revisit chapter 2.

As previously discussed, SVM performance is heavily influenced by the choice of its parameter values, as well by the choice of kernels functions [75]. Regarding the latter issue, there is no scientific rule-of-thumb to follow when deciding which kernel to use because it greatly depends upon the data itself, structure-wise [76], which is why often several kernel functions are tested for the specific data in analysis.

In the implemented prototype, SVM uses the Gaussian kernel (or Radial Basis Function kernel), which is considered, by the scientific community, a good default choice for continuous data [77]. In this set up, two non-trainable parameters, also known as hyperparameters, are computed, C and σ. C parameter controls the complexity of the boundary between support vectors and sigma is a smoothing parameter. As these values are not changed during training, they can be either specified directly by the user, or they can be estimated, through a grid search [78], in which case the values that have lead to better performance are the ones chosen.

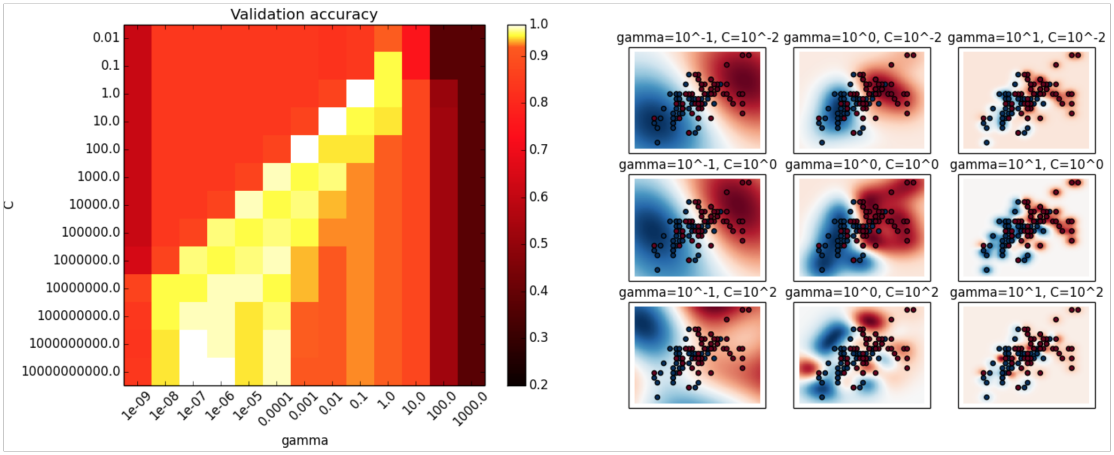

Figure 30: Grid search example: how the values of C and gamma parameters influence the model performance, specifically illustration of SVM with RBF kernel. Iris dataset, using the Scikit-learn tool [79].

In HypE, the system uses leave one-out cross validation (LOOCV) for estimating the best parameter values to use to train the model. Briefly, LOOCV consists on picking one single case as the test set, at a time, and doing it so that every point is used once as a test case. Although LOOCV might be time expensive, which increased with the data size, it is a cross validation technique often used when the data available is limited. The reason for that is that by leaving one point out, at each iteration, it is ensured that the training set is constantly representative of the whole dataset [80].

One of the biggest advantages of LOOCV is that it is an unbiased estimate of the

generalization error. However, the fact that almost all data is used for training, in all iterations, leads to higher variance too, due to the overlap between training sets [81].

Given the fact that diverse subsets are analyzed in the system, it made theoretical sense to use the grid search strategy for computing the "best" parameters, for each dataset.

In practice, it was only specified a range of values of the parameter C, and for each, a σ estimation value was analytically derived. This choice was suggested by the default implementation of the machine learning tool used, and it was kept due to be considered a good solution to control the time expense of the cross validation operation, in case the datasets are to be sizable. Thus, the range values considered were:

- Parameter C: range [0.25, 0.50, 1.00, 2.00, 4.00, 8.00, 16.00, 32.00, 64.00]

After the first pass on the data which aims at finding the best values for the hyperparameters, then a second pass is done by the SVM algorithm, which then models the data with the freshly chosen values of the C and σ parameters.

5.3.3 Model evaluation

In order to assess the potential predictive power of the dataset being analyzed, HypE models the data five different times and compares the performances. The reason why five models are run is to guarantee, to a certain extent, the robustness of the dataset. If four out of five models show a performance no better than random, it can be interpreted as the dataset not containing any distinctive patterns that might be of interest and that that one model probably modeled data noise as predictive patterns.

HypE compares the different models' performances based on the area under the receiver operating characteristic (ROC) curve (AUC) [82]. To simply put it, ROC is a mechanism for visualizing the performance of a binary classifier and AUC is an instrument for summarizing a binary classifier performance.

A binary classifier is one which output can be only one out of two possible classes, meaning that any given input can be classified as either class A or class B. Regarding misclassifications, there are two cases to account for: the true positives and true negatives, which Figure 31 shows how to calculate. Additionally, it also shows some of the most common performance metrics usually computed, namely accuracy, precision and recall, and the F-score.

ROC graphs are two-dimensional graphs that are built from the true positive (TP) and false positive (FP) rates [83]. Specifically, the ROC represents the FP in X axis and the TP on the Y axis, thus illustrating the tradeoff between the true positives and true negatives. Figure 31 shows how to calculate these rates and how does a ROC plot looks like.

Simply stated, AUC corresponds to the area percentage of the ROC plot that is under its curve [85]. A perfect model would have an AUC of 1, as the ROC plot would be a curve that followed the upper left side of the plot, and a model no better

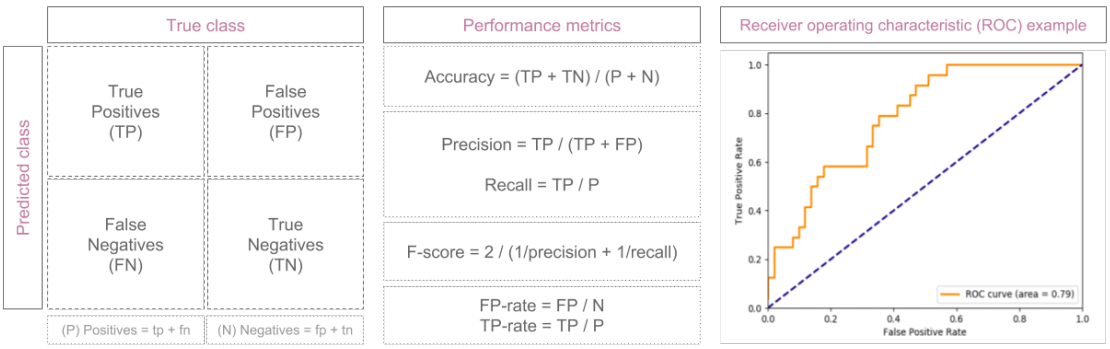

Figure 31: From left to right: (a) binary classifier confusion matrix detailed; (b) several model performance metrics computed from the confusion matrix; (c) example of ROC curve [84].

than random would have an AUC of 0.5, as the ROC would be close to the diagonal of the plot, which is usually plotted for visual reference in Figure 31.

AUC was the performance metric of choice to compare the models' performances, for each dataset being analyzed, mainly do to the fact that it has shown to be a better metric than accuracy, specially when considering unbalanced datasets, meaning datasets in which one class is (much) more common than the other [86]. Given the nature of data used for the development, ICU data, it is expected than some datasets will be indeed unbalanced, for example a dataset regarding a medical condition that is highly deadly.

As it was mentioned before, each dataset is modeled five times, each one with an independent estimation of the C parameter value. Afterwards, the AUC of all runs are compared, as Figure 32 exemplifies. In the prototype implemented, a dataset is considered as containing potentially interesting patterns if the model shows, in at least three out of the five runs, an AUC of value superior of 0.70. This threshold was essentially an educated guess, based on some different subsets of the MIMIC data.

5.4 Hypothesis Embodiment

In the successful case that the data is considered to have interesting predictive patterns, according to the defined thresholds, the next and final step is to somehow show those patterns to the user. The process of presenting them in a human-friendly format corresponds to the embodiment of the hypothesis phase. This phase of HypE is then responsible for representing the patterns found in a way that the user can effortlessly understand them. For example, complex hypotheses involving nested item-value patterns found would be presented in a condensed form such as "in a certain dataset, patients that show an accentuated decrease of hemoglobin, and then an increase of leukocyte values, are much more likely to die".

Considering the intended easy-to-interpret output, decision trees seemed to be a very clear way to quickly understand such patterns and identify different possible scenarios to look out for, and their correspondent outcome (death or surviving) rates, as they have been shown to perform very well in clinical scenarios [87].

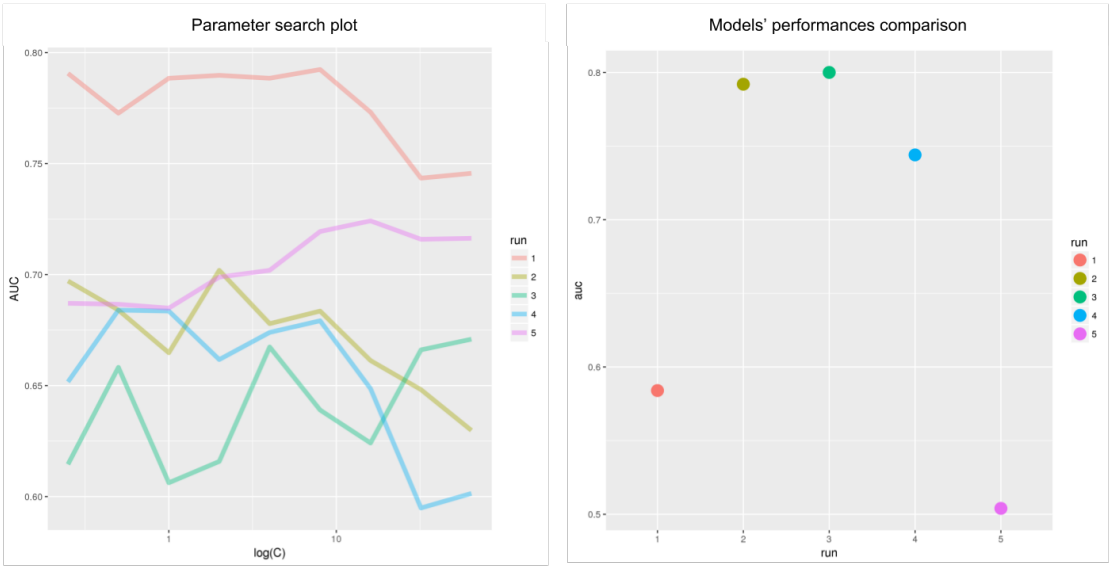

Figure 32: Left picture: parameter tuning search plot - how different values of C influence the overall model performance. Right picture: the model performance values, with the best C parameter estimation, throughout five runs.

Figure 33 exemplifies how the output was implemented in the prototype, and also details the visualization components.

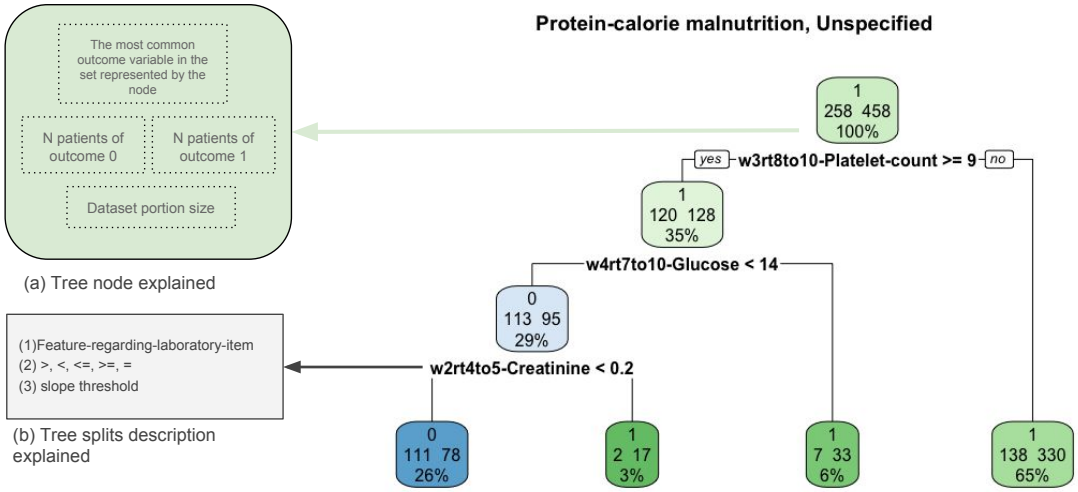

Figure 33: HypE's prototype output visualization by decision trees, for a real MIMIC subset of patients diagnosed with "Protein-calorie malnutrition".

In the prototype, the algorithm chosen to build the DT was the CART [88]. CART is a common implementation of binary decision trees, which means that each node is split into two branches, making the resulting trees easy to understand. In the

prototype, two parameters of CART were specifically defined: the tree complexity parameter and its maximum accepted depth.

In a nutshell, the complexity parameter of a decision tree greatly influences the splits generated, as it enforces that all generated splits must decrease the overall classification error by at least its value. In other words, it can be seen as a built-in pruning that avoids splitting the data into sets that will bring the resulting tree no performance advantage. HypE's uses a complexity parameter of value 0.02, which means that each split should increase the DT's performance in at least 2%.

The self-explanatory parameter for the maximum accepted depth of the tree, limits the amount of node splits allowed. The deeper the trees, the harder to interpret the results, which is why HypE is limited to trees with three levels of nodes (the root counts as level zero).

The decision trees outputed, however, differ from the most common used DT trees, given the nature of the data being used to build them. Note that the feature data consists on the slope between any two points in time, which means that the tree split values will be slopes as well. Having acknowledged this, the interpretation is not as easy as checking some patient current value for some laboratory measurement and it does raise the question of how to categorize the slopes into natural language conclusions. For example, starting from what slope value is an increase considered "significant" as in "item A showed a significant decrease".

Nevertheless, it is now worth reminding that the contribution of this system is not in the use of certain methods for certain tasks, but in the pipeline of operations that allow for the identification of data patterns by automatic analyzing diverse raw laboratory data. Hence, the implemented prototype uses decision trees (DT) algorithm for generating an easy-to-read output, but any other computational method or visualization tool can be used.

6 Results

Given the nature of HypE's goal to automatic mine data sets in search for biomedical relevant phenomena, reporting results requires manual selection of examples, specially if we want to present cases in which no hypothesis was found.

Considering that the system aims at precisely directing the user's attention to datasets that might have been missed otherwise, there is no metric that can be used to evaluate the performance of the system implemented. Instead, some examples of what HypE outputted will be presented.

6.1 Diagnoses Subsets

In this prototype, all subsets analyzed were diagnosis-based, by which we mean that they were sliced out of the original data, by a certain diagnosis's patients cohort, regardless of any additional diagnosis. For instance, one diagnosis present in the MIMIC database is *chronic diastolic heart failure* (ICD9-Code *42832*), and so the correspondent subset would consist on all patients associated with this diagnosis.

There are two essential issues to consider when modeling data in HypE: the amount of the data being analyzed, at once, and the balance of the output variable.

Generically speaking, bigger sets of data are expected to train better predictive models which extrapolates that HypE is more likely to find interesting phenomena the bigger the sets are. For instance, some diagnosis are so rare that only a few cases are reported in MIMIC database, making any predictive analysis meaningless. Figure 34 shows the subset sizes distribution; the figure is spitted into three groups because the amount of diagnosis that had less than 20 patients is much bigger than all the rest. For example, drug-related admissions are quite rare in MIMIC, even more so because there are over 10 different ICD9 code for different drug type and frequency use.

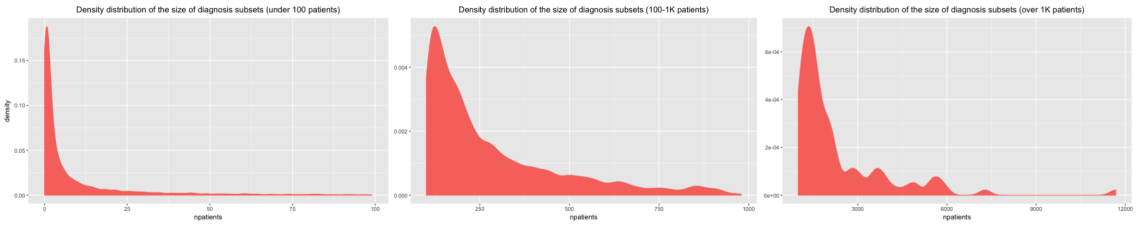

Figure 34: Density distribution plot of the size of all available subsets, per "cluster": dataset associated with less than 100 patients (left), from 101 to 999 patients (middle), 1000 patients or more(right).

With respect to the survival ratio, the amount of surviving *versus* death patients within a data subset, it is of great importance as it is the outcome variable considered in the prototype that was tested. Too unbalanced subsets are not good for modeling, as the models are likely to be biased once the data clearly has much more of one outcome class than the other. Figure 35 shows that are many diagnosis sets with a balanced outcome ratio, although it does show two high density areas in the extreme of the plot, which represent both the most and least lethal diagnosis.

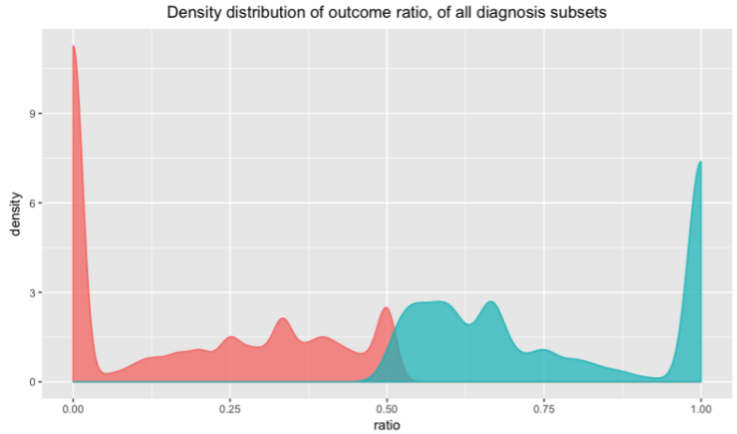

Figure 35: Density distribution of the outcome ratio of all datasets. Red color represents a low death ration (more subjects alive), and the blue a high death ratio (more subjects dead).

6.2 Hypothesis extracted

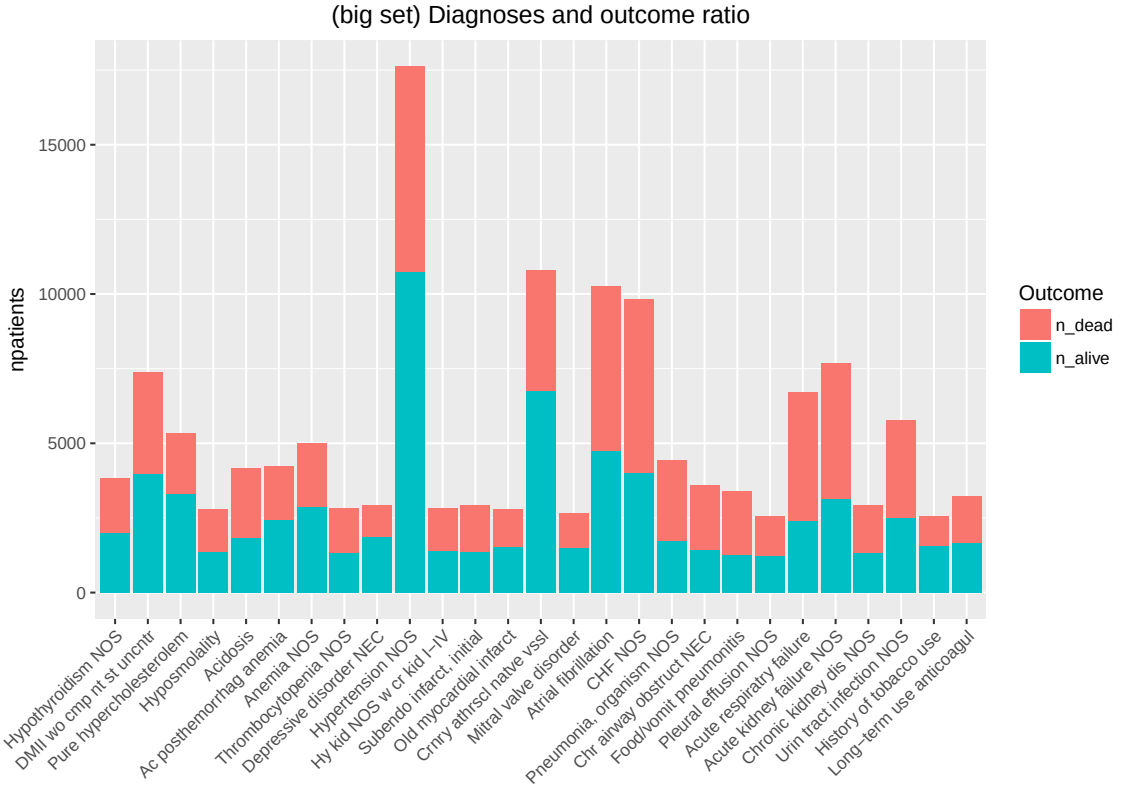

Figure 36: MIMIC diagnosis with most patients associated, and the correspondent outcome (alive/dead) ratio. The diagnosis descriptions correspond to the official ICD9 codes short description.

This section details a few diverse examples that show both the potential and limitations of HypE, mainly focusing on the output embodiment. It is worth mentioning

that as no medical physicians were involved in the development of this project, there was no scientific crosschecking of research with the results that are presented in this chapter.

Figure 36 shows the most common diagnostics in the data, as well as the correspondent survival ratios. As it can be seen, *"unspecified essential hypertension"* is undoubtedly the biggest diagnostic subset and shows a good survival ratio, but the diagnosis itself states its non-specificity, which can make any hypothesis found even harder to interpret as a promising. Hence, choosing clearer diagnostic subsets eases the result analysis, as the results can be, for example, crossed with that diagnosis related research and verify if the hypotheses make any biomedical sense, at all. Out of the biggest sets in the data, we did analyze the sets of *Coronary atherosclerosis of native coronary artery* and *arterial fibrillation*, but unfortunately no hypothesis were found. A plausible reason for this might be these sets heterogeneity, in regards to other diagnosis the patients may have, for example.

With slightly less patients, we present the results for the *mitral valve disorder*, *depressive disorder* and *bacteremia* cohorts. Figure 37 details the patient sample size during the previously described feature engineering phase, for all three diagnoses sets that were analyzed, and which results are presented in the next sections.

Diagnosis description (ICD9 short description)	MIMIC (#patients)	After data subsetting (#patients)	After feature selection (#patients)
Mitral valve disorder	2651	1596	1057
Depressive disorder NEC	2926	1408	868
Bacteremia	1321	940	696

Figure 37: Patient sample size (how many patients), in the original MIMIC database, after the data sub-setting (time-restriction requirements), and in the feature data (final set used for analysis).

The reason for choosing these sets was essentially the different nature of the diagnosis themselves. In non-professional terms, mitral valve disease is when the mitral valve (located between the left heart chambers) does not work properly. A quick search online shows evidence that complications from the disease are known causes for the patient to end up in an ICU. On the other hand, depressive disorder is a mental disorder, which, by itself, is unlikely to be the primal reason for the patient to be admitted into an ICU, but rather likely to be a second diagnosis. Finally, bacteremia is, in short, the presence of bacteria in blood, which can lead to sepsis and death. Because a patient can contract bacteremia within hospital, i.e. during their hospital stay, it was considered of interesting to look into the extracted hypothesis.

6.2.1 Mitral Valve Disorder

Mitral valve disorder has been showed to be associated with atrial fibrillation, which is one of the main diagnosis present in the database, as Figure 36 shows. However, in this case, HypE did find an hypothesis worth outputting.

The resulting DT is an example of the shortest possible outputted, with one node level only. Both tree leafs show the presence of both outcomes, which is not ideal, even though excepted, specially with such a non-deep tree.

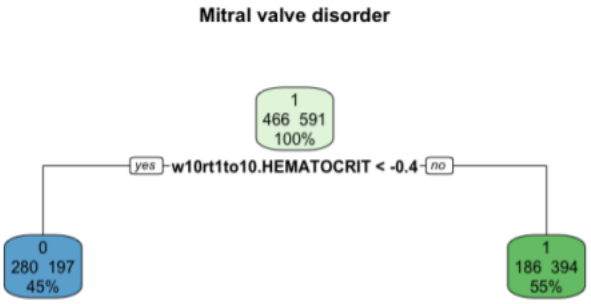

Figure 38: Decision trees embodying the hypotheses identified in the sets of the mitral valve diagnose.

6.2.2 Depressive disorder

Unlikely the previous example, the depressive disorder tree,, has more node splits, which usually means that the leafs are more homogeneous, outcome wise, even though it is not exactly the case here.

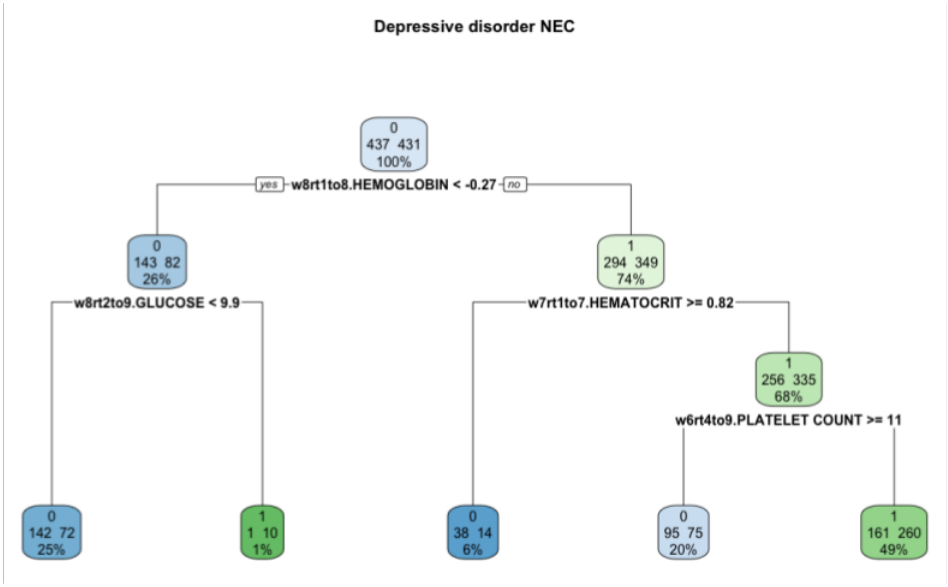

Figure 39: Decision trees embodying the hypotheses identified in the set of depressive disorders diagnose.

Considering the database used consists on patients admitted into intensive care units, it is safe to say that mental disorders, such as depression, are unlikely to be the main reason for the hospital stay. Instead, it is highly likely that depression is

just another diagnosis that the patient is associated with. For instance, depression can be a result of a long illness, or for example given the prospect of near death [89]. Thus, the cohort data used in this example is most likely highly heterogeneous, specially regarding the conditions that may have lead to the patients extreme health conditions and consequent admission into an ICU.

6.2.3 Bacteremia

As it was mentioned before, bacteremia is a condition that can be contracted during hospital stay, and lead to sepsis and even death. In-hospital contracted conditions are oftentimes studied and tools like HypE could be of great use to try to find patterns of infection spreading that may have not yet been identified by physicians. Thus, there was a certain scientific curiosity to see what Hype would output with such dataset.

Surprisingly, some hypothesis were indeed outputted. Figure 40 presents the results for the dataset of diagnosis *bacteremia*. In short, bacteremia refers to the presence of bacteria in blood. Interestingly, looking at the outputted tree, we have blood components which evolution is commonly tracked in a case of infections, in general, and bicarbonate can be used in the treatment of bacteremia, as it is mentioned in [90].

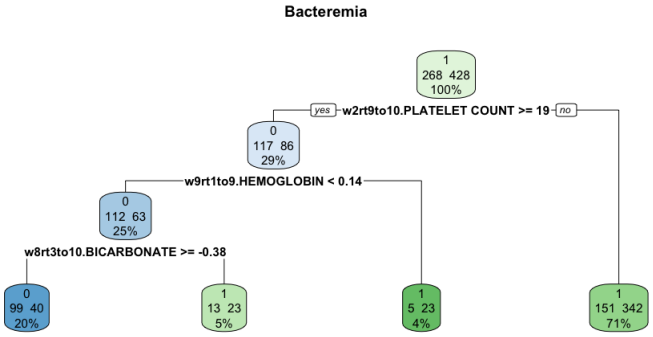

Figure 40: Decision trees embodying the hypotheses identified in the sets of bacteremia.

7 Discussion

The HypE system aimed at developing a data mining tool capable of automatically mine big biomedical datasets, from different angles (meaning datasets), and identify potentially interesting data patterns that seemed to be worth looking into. As such, HypE was envisaged to be a clinical decision support system, or at least to be integrated into one.

The implemented prototype was a partial implementation of the whole system, as the main goal of this project was HypE's proof of concept. As such, operations like sub-setting the data for analysis, how that data is analyzed, and how the results are visualized are currently limited to the implementation choices we've made. Nevertheless, as it was mentioned before, the contribution of this work is in the pipeline, and not the specific methods or tools used for operation A or B.

Figure 41 illustrates the prototype complete pipeline, and details the implementations choices for each step.

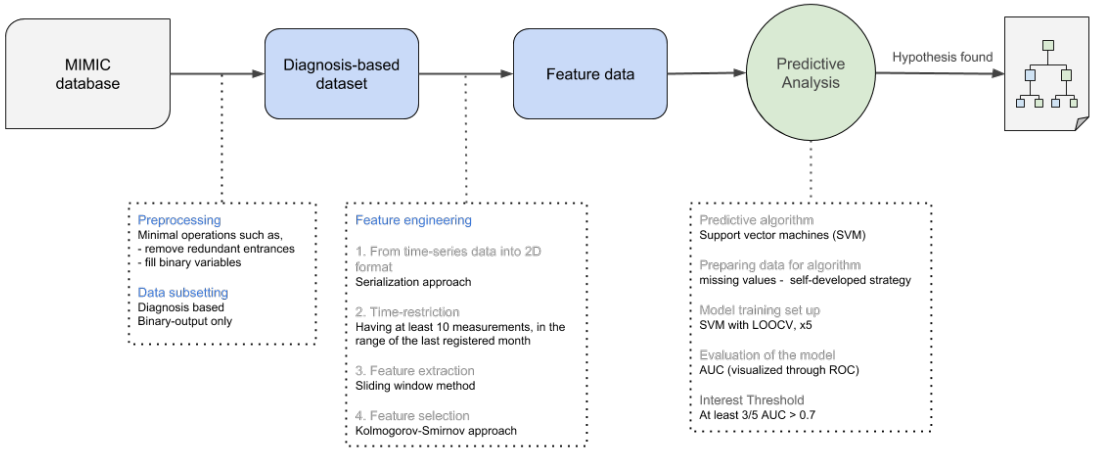

Figure 41: The prototype pipeline, and specifications of operations and implementation choices.

The stage designated as feature engineering corresponds to a sequence of steps that manipulates the data into the format needed for the predictive analysis. Cross checking the results with the prototype pipeline, it is clear that the feature engineering phase leads to a reduction of the data available for analysis, which was expected, given the irregular nature of clinical data, time-wise, i.e., some patients might have several laboratory items being tracked through time while others in a similar medical condition might not have those same data.

Furthermore, the choice of support vector machines algorithm for the predictive analysis also involves sacrificing some more data, given the version used does not support missing values. The strategy used for this step was developed specifically for this project, and its goal was to diminish the amount of samples to be dropped from the dataset, and prioritizing the exclusion of features that were "too" sparse, according to the defined threshold. The alternative considered during the development was to

estimate the missing data values, by some imputation method, which would mean that data being analyzed would not be entirely "real".

Regarding the serialization of the time-series data, one conceptual change that could be changed would be to consider different admissions, as different "samples". For example, a patient that has been twice in the ICU, long enough for it to have data meeting the requirements for analysis, could be split into two different samples, for example.

Lastly, the outputted decision trees unveiled a problem with using slope values as features, which is the interpretation of the actual output. In fact, the hypothesis embodiment step would likely be *Achilles heel* of the prototype. One important aim of the system was to package the medical phenomena in a way that would be easy to interpret, and that is not the case with "slope" values. In theory, it made sense to think of the time evolution of an item as a straight line and that the slope would indicate whether it decrease, increased or didn't change. However, in practice, the DT splits are conditions such as "slope higher that 0.34", which include anything from so very minimal increase change to very rapid increase. Thus, it is acceptable to say that DT might not be the best choices for outputting this type of information. One strategy could be to firstly convert the continuous slope values into categories, like "slight increase" or "rapid increase" and then use some other output format to clearly represent such pattern.

Still in the topic of the output interpretative value, one issue that was highlighted by the DT examples was that the slopes do not contain any information regarding whether they represent values that were outside the recommended value range. In other words, it might be important to understand if the increase or decrease of, for example, hemoglobin, is happening within the clinical references, because if not, such variation might have a different meaning. Therefore, finding a strategy to include this information in the analyzes would be of great usefulness.

7.1 System limitations

As expected with any proof of concept project, the system does have several limitations, regarding the pipeline itself. This section describes them and how they could be overcome in the future.

Regarding the data subsetting, the diagnosis-based approach was a simple method that would allow us to divide the database into many different sets, which was an advantage in the development phase. However, in future versions of the system, other logic for data splitting could be used. For instance, ICD9 has many different codes that relate to similar scenarios, of example *cocaine dependence, unspecified (ICD9 code: 304.20)*, *cocaine dependence, continuous (ICD9 code: 304.21)*, *cocaine dependence, episodic (ICD9 code: 304.22)*, *cocaine dependence, in remission (ICD9 code: 304.23)*. It could be interesting to merge them into one single cohort before analyzing it for medical phenomena, or maybe filter some other cohorts by age, or other conditions.

The choice of supporting only binary target variables was mainly to match the intended output, as the decision trees are used to visualize binary numerical data

only. Nevertheless, even with such a simple output type scenario, more complex cohorts could be already analyzed, for example, considering other diagnosis, or specific condition, as output, in which case an extra layer of analysis would be needed, in order to consider the multiple comparisons problem[91].

Still concerning the data subsetting, maybe the biggest limitation of HypE, as of now, is the exclusive support of numerical data. We do believe that categorical data may contain lots of relevant information, maybe more so that numerical data, which is usually what is the base of EHR data analysis, but it would require heavier data processing and different choices for the predictive analysis algorithms.

Technically, HypE was implemented in R[92], making use of several different packages, for particular operations, and it is completely functional on any data that is loaded into the system.

However, the development of the prototype did not take into consideration issues like performance and efficiency, which means that the process from input to output might take longer than ideal. Parallelization was not considered, which means that the pipeline runs once at a time, not allowing for multiple datasets analyses. Nonetheless, making it parallel-proof would not require major changes.

7.2 Future Work

The biggest change we would recommend going forward would likely be the programming language, as a system with the characteristics and goals of HypE would certainly benefit from fast performance, of which R is not a particular example of.

In the case of wanting to actually test out HypE in a real world scenario, the biggest challenge would very likely be whether or not to consider it as a standalone tool, or integrate it into some other system. In the first scenario, and considering the current performance was not an issue, the prototype as it is could definitely mine any database, requiring however the data subsetting process to be customized for that database. In the second scenario, it would anyhow depend upon the integration of it into another existent system, which is a talk of which the complexity of we cannot access without knowing the system in question.

8 Conclusion

First of all, it is important to recap what HypE's purpose is and what the expectations of this proof of concept project were.

The hypothesis engine was a system that was envisaged to automatically mine electronic health records (EHR) data and output cohorts that seemed to have some patterns that could make researches or physicians want to look into.

The motivation essentially came from the belief that patients with certain diagnoses might have distinct indications of either survival or death, medically-wise. As such, in those case, it would make theoretical sense to expect HypE to find hypothesis that would be easily associated with the diagnoses in research regarding precisely those conditions.

It is important to understand that HypE does not intend to replace medical research or even to find unknown groundbreaking connections/correlations/causality within conditions, although theoretically it could. Instead, HypE wants to support the medical staff by signalizing biomedical phenomena, as soon as it is possible, ideally before the doctor himself realizes it.

Naturally, the developed prototype was a somewhat limited version of what we believe a system like HypE can do, which has necessarily adjusted our expectations. On one hand, the data subsets analyzed were diagnosis-based, which makes sense on paper, but when considering an EHR database, and the fact that there are lots of different ICD9 codes for similar diagnosis, would make lots of subsets too short for analysis when they could be analyzed as a single cohort.

On the other hand, the publicly available database used as a data source, MIMIC, consists on data from patients admitted into ICU units. From the clinical perspective, it narrows down the potential interpretation of whatever hypothesis might be found, considering these are patients in extreme health conditions. In other words, the additional fact that all the data comes from subjects in severe condition is likely to make the data noisy, to some extent, which is why the results were not expected to be in a certain way.

Looking at the results, we do believe that automatically analyze big datasets in search for cohorts of data the show interesting characteristics is definitely, not only valuable, but possible. Tools like HypE can hint to things that might be missed otherwise, and could be used in cooperative or research environments.

Since the beginning of this project, interesting research has been published regarding the mining on EHR data, and using MIMIC database as well, and exploring more recent ML architectures, such deep learning, which has achieved very promising results [45, 93].

Biomedical data is as messy as it is rich in information, which is why translating such information into knowledge is of great value, as useful data is much more worthy than non useful data.

References

[1] Andrea D Weston and Leroy Hood. Systems biology, proteomics, and the future of health care: toward predictive, preventative, and personalized medicine. *Journal of proteome research*, 3(2):179–196, 2004.

[2] Andrew Miles, Michael Loughlin, and Andreas Polychronis. Evidence-based healthcare, clinical knowledge and the rise of personalised medicine. *Journal of Evaluation in Clinical Practice*, 14(5):621–649, 2008.

[3] Lynda Chin, Jannik N Andersen, and P Andrew Futreal. Cancer genomics: from discovery science to personalized medicine. *Nature medicine*, 17(3):297–303, 2011.

[4] John F Mccarthy, Kenneth A Marx, Patrick E Hoffman, Alexander G Gee, Philip O'neil, M L Ujwal, and John Hotchkiss. Applications of machine learning and high-dimensional visualization in cancer detection, diagnosis, and management. *Annals of the New York Academy of Sciences*, 1020(1):239–262, 2004.

[5] William Hersh. A stimulus to define informatics and health information technology. *BMC Medical Informatics and Decision Making*, 9(1):24, 2009.

[6] Mark A Musen, Blackford Middleton, and Robert A Greenes. Clinical decision-support systems. In *Biomedical informatics*, pages 643–674. Springer, 2014.

[7] Eta S Berner and Tonya J La Lande. Overview of clinical decision support systems. In *Clinical decision support systems*, pages 1–17. Springer, 2016.

[8] Stuart Russell, Peter Norvig, and Artificial Intelligence. A modern approach. *Artificial Intelligence. Prentice-Hall, Egnlewood Cliffs*, 25:27, 1995.

[9] John McCarthy. What is artificial intelligence. *URL: http://www-formal. stanford. edu/jmc/whatisai. html*, page 38, 2007.

[10] Michael Wooldridge and Nicholas R Jennings. Intelligent agents: Theory and practice. *The knowledge engineering review*, 10(02):115–152, 1995.

[11] Ethem Alpaydin. *Introduction to machine learning*. MIT press, 2014.

[12] Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

[13] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. MIT press, 2012.

[14] R Chitra and V Seenivasagam. Heart disease prediction system using supervised learning classifier. *Bonfring International Journal of Software Engineering and Soft Computing*, 3(1):1, 2013.

[15] Rima Kaddurah-Daouk, Bruce S Kristal, and Richard M Weinshilboum. Metabolomics: a global biochemical approach to drug response and disease. *Annu. Rev. Pharmacol. Toxicol.*, 48:653–683, 2008.

[16] Sebastian Raschka. *Python machine learning*. Packt Publishing Ltd, 2015.

[17] Pedro Domingos. A unified bias-variance decomposition. In *Proceedings of 17th International Conference on Machine Learning*, pages 231–238, 2000.

[18] Sylvain Arlot, Alain Celisse, et al. A survey of cross-validation procedures for model selection. *Statistics surveys*, 4:40–79, 2010.

[19] Manuel Fernández-Delgado, Eva Cernadas, Senén Barro, and Dinani Amorim. Do we need hundreds of classifiers to solve real world classification problems. *J. Mach. Learn. Res*, 15(1):3133–3181, 2014.

[20] Hyeran Byun and Seong-Whan Lee. Applications of support vector machines for pattern recognition: A survey. *Pattern recognition with support vector machines*, pages 571–591, 2002.

[21] Christopher JC Burges. A tutorial on support vector machines for pattern recognition. *Data mining and knowledge discovery*, 2(2):121–167, 1998.

[22] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An introduction to statistical learning*, volume 112. Springer, 2013.

[23] Kristin P Bennett and Colin Campbell. Support vector machines: hype or hallelujah? *Acm Sigkdd Explorations Newsletter*, 2(2):1–13, 2000.

[24] Martin Hofmann. Support vector machines-kernels and the kernel trick. *An elaboration for the Hauptseminar Reading Club SVM*, 2006.

[25] John Shawe-Taylor and Nello Cristianini. *Kernel methods for pattern analysis*. Cambridge university press, 2004.

[26] Yuichi Motai. Kernel association for classification and prediction: A survey. *IEEE transactions on neural networks and learning systems*, 26(2):208–223, 2015.

[27] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[28] Pedro Larranaga, Borja Calvo, Roberto Santana, Concha Bielza, Josu Galdiano, Iñaki Inza, José A Lozano, Rubén Armañanzas, Guzmán Santafé, Aritz Pérez, et al. Machine learning in bioinformatics. *Briefings in bioinformatics*, pages 86–112, 2006.

[29] Joseph A Cruz and David S Wishart. Applications of machine learning in cancer prediction and prognosis. *Cancer informatics*, 2:59, 2006.

[30] Carl Kingsford and Steven L Salzberg. What are decision trees? *Nature biotechnology*, 26(9):1011–1013, 2008.

[31] Lior Rokach and Oded Maimon. Top-down induction of decision trees classifiers-a survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 35(4):476–487, 2005.

[32] M Möhlig, A Flöter, J Spranger, MO Weickert, T Schill, HW Schlösser, G Brabant, AFH Pfeiffer, J Selbig, and C Schöfl. Predicting impaired glucose metabolism in women with polycystic ovary syndrome by decision tree modelling. *Diabetologia*, 49(11):2572–2579, 2006.

[33] Rich Caruana, Nikos Karampatziakis, and Ainur Yessenalina. An empirical evaluation of supervised learning in high dimensions. In *Proceedings of the 25th international conference on Machine learning*, pages 96–103. ACM, 2008.

[34] Miles N Wernick, Yongyi Yang, Jovan G Brankov, Grigori Yourganov, and Stephen C Strother. Machine learning in medical imaging. *IEEE signal processing magazine*, 27(4):25–38, 2010.

[35] Kunio Doi. Diagnostic imaging over the last 50 years: research and development in medical imaging science and technology. *Physics in Medicine & Biology*, 51(13):R5, 2006.

[36] Kunio Doi. Current status and future potential of computer-aided diagnosis in medical imaging. *The British journal of radiology*, 78(suppl_1):s3–s19, 2005.

[37] Kunio Doi. Computer-aided diagnosis in medical imaging: historical review, current status and future potential. *Computerized medical imaging and graphics*, 31(4-5):198–211, 2007.

[38] Meherwar Fatima and Maruf Pasha. Survey of machine learning algorithms for disease diagnostic. *Journal of Intelligent Learning Systems and Applications*, 9(01):1, 2017.

[39] Maryam M Najafabadi, Flavio Villanustre, Taghi M Khoshgoftaar, Naeem Seliya, Randall Wald, and Edin Muharemagic. Deep learning applications and challenges in big data analytics. *Journal of Big Data*, 2(1):1, 2015.

[40] Geert Litjens, Thijs Kooi, Babak Ehteshami Bejnordi, Arnaud Arindra Adiyoso Setio, Francesco Ciompi, Mohsen Ghafoorian, Jeroen AWM van der Laak, Bram Van Ginneken, and Clara I Sánchez. A survey on deep learning in medical image analysis. *Medical image analysis*, 42:60–88, 2017.

[41] Hayit Greenspan, Bram Van Ginneken, and Ronald M Summers. Guest editorial deep learning in medical imaging: Overview and future promise of an exciting new technique. *IEEE Transactions on Medical Imaging*, 35(5):1153–1159, 2016.

[42] June-Goo Lee, Sanghoon Jun, Young-Won Cho, Hyunna Lee, Guk Bae Kim, Joon Beom Seo, and Namkug Kim. Deep learning in medical imaging: general overview. *Korean journal of radiology*, 18(4):570–584, 2017.

[43] Jonghoon Kim, Jisu Hong, Hyunjin Park, Jonghoon Kim, Jisu Hong, and Hyunjin Park. Prospects of deep learning for medical imaging. *Precision and Future Medicine*, 2(2):37–52, 2018.

[44] Steven Lemm, Benjamin Blankertz, Thorsten Dickhaus, and Klaus-Robert Müller. Introduction to machine learning for brain imaging. *Neuroimage*, 56(2):387–399, 2011.

[45] Kristiina Häyrinen, Kaija Saranto, and Pirkko Nykänen. Definition, structure, content, use and impacts of electronic health records: a review of the research literature. *International journal of medical informatics*, 77(5):291–304, 2008.

[46] Adam Hedgecoe. *The politics of personalised medicine: Pharmacogenetics in the clinic*. Cambridge University Press, 2004.

[47] Xianwen Ren, Yong Wang, Xiang-Sun Zhang, and Qi Jin. ipcc: a novel feature extraction method for accurate disease class discovery and prediction. *Nucleic acids research*, page gkt343, 2013.

[48] Qinbao Song, Jingjie Ni, and Guangtao Wang. A fast clustering-based feature subset selection algorithm for high-dimensional data. *IEEE transactions on knowledge and data engineering*, 25(1):1–14, 2013.

[49] Alok Sharma, Seiya Imoto, and Satoru Miyano. A top-r feature selection algorithm for microarray gene expression data. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, 9(3):754–764, 2012.

[50] Konstantina Kourou, Themis P Exarchos, Konstantinos P Exarchos, Michalis V Karamouzis, and Dimitrios I Fotiadis. Machine learning applications in cancer prognosis and prediction. *Computational and structural biotechnology journal*, 13:8–17, 2015.

[51] Filippo Amato, Alberto López, Eladia María Peña-Méndez, Petr Vaňhara, Aleš Hampl, and Josef Havel. Artificial neural networks in medical diagnosis, 2013.

[52] Ahmad Taher Azar and Shereen M El-Metwally. Decision tree classifiers for automated medical diagnosis. *Neural Computing and Applications*, 23(7-8):2387–2403, 2013.

[53] Li-Yeh Chuang, Kuo-Chuan Wu, Hsueh-Wei Chang, and Cheng-Hong Yang. Support vector machine-based prediction for oral cancer using four snps in dna repair genes. In *Proceedings of the International MultiConference of Engineers and Computer Scientists, Hong-Kong*. Citeseer, 2011.

[54] Mehmet Fatih Akay. Support vector machines combined with feature selection for breast cancer diagnosis. *Expert systems with applications*, 36(2):3240–3247, 2009.

[55] Peter Lucas. Bayesian analysis, pattern analysis, and data mining in health care. *Current opinion in critical care*, 10(5):399–403, 2004.

[56] Peter JF Lucas, Linda C van der Gaag, and Ameen Abu-Hanna. Bayesian networks in biomedicine and health-care. *Artificial intelligence in medicine*, 30(3):201–214, 2004.

[57] Ritesh Singh, Keshab Mukhopadhyay, et al. Survival analysis in clinical trials: Basics and must know areas. *Perspectives in clinical research*, 2(4):145, 2011.

[58] David Ost, Judith Goldberg, Linda Rolnitzky, and William N Rom. Survival after surgery in stage ia and ib non–small cell lung cancer. *American journal of respiratory and critical care medicine*, 177(5):516–523, 2008.

[59] Wanpracha Art Chaovalitwongse, Ya-Ju Fan, and Rajesh C Sachdeo. On the time series k-nearest neighbor classification of abnormal brain activity. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 37(6):1005–1016, 2007.

[60] Yuu Yamada, Einoshin Suzuki, Hideto Yokoi, and Katsuhiko Takabayashi. Decision-tree induction from time-series data based on a standard-example split test. In *ICML*, volume 3, pages 840–847, 2003.

[61] John W Tukey. Some thoughts on clinical trials, especially problems of multiplicity. *Science*, 198(4318):679–684, 1977.

[62] Stuart J Pocock, Michael D Hughes, and Robert J Lee. Statistical problems in the reporting of clinical trials. *New England journal of medicine*, 317(7):426–432, 1987.

[63] Harold T Shapiro and Eric M Meslin. Ethical issues in the design and conduct of clinical trials in developing countries, 2001.

[64] Vijay Huddar, Bapu Koundinya Desiraju, Vaibhav Rajan, Sakyajit Bhattacharya, Shourya Roy, and Chandan K Reddy. Predicting complications in critical care using heterogeneous clinical data. *IEEE Access*, 4:7988–8001, 2016.

[65] Alistair EW Johnson, Mohammad M Ghassemi, Shamim Nemati, Katherine E Niehaus, David A Clifton, and Gari D Clifford. Machine learning and decision support in critical care. *Proceedings of the IEEE*, 104(2):444–466, 2016.

[66] Alistair EW Johnson, Tom J Pollard, Lu Shen, Li-wei H Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. Mimic-iii, a freely accessible critical care database. *Scientific data*, 3, 2016.

[67] Gari D Clifford, Daniel J Scott, Mauricio Villarroel, et al. User guide and documentation for the mimic ii database. *MIMIC-II database version*, 2(95), 2009.

[68] Andre Dejam, Brian E Malley, Mengling Feng, Federico Cismondi, Shinhyuk Park, Saira Samani, Zahra Aziz Samani, Duane S Pinto, and Leo Anthony Celi. The effect of age and clinical circumstances on the outcome of red blood cell transfusion in critically ill patients. *Critical care*, 18(4):487, 2014.

[69] Lior Fuchs, Victor Novack, Stuart McLennan, Leo Anthony Celi, Yael Baumfeld, Shinhyuk Park, Michael D Howell, and Daniel S Talmor. Trends in severity of illness on icu admission and mortality among the elderly. *PloS one*, 9(4):e93234, 2014.

[70] World Health Organization and Practice Management Information Corporation. *ICD-9-CM: International Classification of Diseases, 9th Revision: Clinical Modification*, volume 1. PMIC (Practice Management Information Corporation), 1998.

[71] John H Drew, Andrew G Glen, and Lawrence M Leemis. Computing the cumulative distribution function of the kolmogorov–smirnov statistic. *Computational statistics & data analysis*, 34(1):1–15, 2000.

[72] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[73] A Rogier T Donders, Geert JMG Van Der Heijden, Theo Stijnen, and Karel GM Moons. A gentle introduction to imputation of missing values. *Journal of clinical epidemiology*, 59(10):1087–1091, 2006.

[74] Joel Grus. *Data science from scratch: First principles with Python*. " O'Reilly Media, Inc.", 2015.

[75] Alex J Smola, Bernhard Schölkopf, and Klaus-Robert Müller. The connection between regularization operators and support vector kernels. *Neural networks*, 11(4):637–649, 1998.

[76] Shun-ichi Amari and Si Wu. Improving support vector machine classifiers by modifying kernel functions. *Neural Networks*, 12(6):783–789, 1999.

[77] Shawkat Ali and Kate A Smith-Miles. A meta-learning approach to automatic kernel selection for support vector machines. *Neurocomputing*, 70(1):173–186, 2006.

[78] Chih-Wei Hsu, Chih-Chung Chang, Chih-Jen Lin, et al. A practical guide to support vector classification. 2003.

[79] James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(Feb):281–305, 2012.

[80] Ron Kohavi et al. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai*, volume 14, pages 1137–1145. Montreal, Canada, 1995.

[81] Sudhir Varma and Richard Simon. Bias in error estimation when using cross-validation for model selection. *BMC bioinformatics*, 7(1):91, 2006.

[82] James A Hanley and Barbara J McNeil. The meaning and use of the area under a receiver operating characteristic (roc) curve. *Radiology*, 143(1):29–36, 1982.

[83] Tom Fawcett. An introduction to roc analysis. *Pattern recognition letters*, 27(8):861–874, 2006.

[84] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830, 2011.

[85] Andrew P Bradley. The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern recognition*, 30(7):1145–1159, 1997.

[86] Charles X Ling, Jin Huang, Harry Zhang, et al. Auc: a statistically consistent and more discriminating measure than accuracy. In *IJCAI*, volume 3, pages 519–524, 2003.

[87] Johannes Mair, Bernd Puschendorf, Jörn Smidt, Peter Lechleitner, and Franz Dienstl. A decision tree for the early diagnosis of acute myocardial infarction in nontraumatic chest pain patients at hospital admission. *Chest*, 108(6):1502–1509, 1995.

[88] Terry M Therneau, Elizabeth J Atkinson, et al. An introduction to recursive partitioning using the rpart routines. Technical report, Technical report Mayo Foundation, 1997.

[89] Alexi A Wright, Baohui Zhang, Alaka Ray, Jennifer W Mack, Elizabeth Trice, Tracy Balboni, Susan L Mitchell, Vicki A Jackson, Susan D Block, Paul K Maciejewski, et al. Associations between end-of-life discussions, patient mental health, medical care near death, and caregiver bereavement adjustment. *Jama*, 300(14):1665–1673, 2008.

[90] R Phillip Dellinger, Jean M Carlet, Henry Masur, Herwig Gerlach, Thierry Calandra, Jonathan Cohen, Juan Gea-Banacloche, Didier Keh, John C Marshall, Margaret M Parker, et al. Surviving sepsis campaign guidelines for management of severe sepsis and septic shock. *Intensive care medicine*, 30(4):536–555, 2004.

[91] Yoav Benjamini. Simultaneous and selective inference: Current successes and future challenges. *Biometrical Journal*, 52(6):708–721, 2010.

[92] R Core Team et al. R: A language and environment for statistical computing. 2013.

[93] Benjamin Shickel, Patrick Tighe, Azra Bihorac, and Parisa Rashidi. Deep ehr: a survey of recent advances in deep learning techniques for electronic health record (ehr) analysis. *arXiv preprint arXiv:1706.03446*, 2017.