Universitat Autònoma de Barcelona

Deep Learning Solutions for Clinical Data Challenges in End-Stage Renal Disease

Edwar Hernando Macias Toro

Universitat Autònoma de Barcelona

Department de Telecomunicacions i Enginyeria de Sistemes
Wireless Information Networking (WIN) Group

Ph.D Dissertation

Deep Learning Solutions for Clinical Data Challenges in End-Stage Renal Disease

Edwar Hernando Macias Toro

PhD program in Electronic and Telecommunication Engineering

Supervisors Antoni Morell Perez, Javier Serrano Garcia and Jose Lopez Vicario

April, 2022

Abstract

In the deep learning (DL) era, it is possible to generate learning models that exploit complex relationships from data. In a clinical setting, integrating DL techniques could accelerate the paradigm shift from evidence-based medicine to data-driven medicine to support medical decisions. The DL methodologies may address challenges related to improving predictive models, imputation of missing information and issues related to the volume of information in learning tasks. We aim to provide DL-based solutions to such challenges.

First, we use machine learning techniques to address the mortality of patients in end-stage renal disease (ESRD). Features are found automatically and then compared with groups of variables chosen by expert staff. Specialized DL models are used to exploit temporal dependencies in the data. The integration of a DL approach made it possible to improve the learning models for this pathology significantly. It also explored how the features encountered automatically presented perform for the learning models.

Then, we propose an alternative that integrates multiple imputation with the average latent representation, the so-called average code. The reconstruction significantly improves. This approach is validated on four clinical datasets, emphasizing the dataset related to the mortality of patients with acute kidney disease (AKI). The proposed method was evaluated on two other datasets widely used in the literature as benchmarks, and presented a better reconstruction capacity in most of the proposed scenarios.

Finally, low volume and the class imbalance in ESRD is addressed. We offer a transfer learning solution. It consists of two approaches for increasing samples and the feature space for ESRD data. Latent spaces of autoencoders as an information bridge between target and source domain. AKI data is used as the source domain. As a result, it is obtained that the proposed mechanisms individually improve the prediction models, but when they are combined sequentially, they suppose a much more significant improvement.

Acknowledgement

At the end of this journey, I look back and see all the experiences and lessons that the people who have been part of it have left me. First of all, I would like to thank my directors Dr. Antoni Morell, Dr. Javier Serrano and Dr. José López Vicario for their constant support, dedication and teachings. I would also like to thank Dr. José Ibeas for his willingness to share his experience in the clinical setting.

I would like to thank my colleagues who have become good friends, Guillem Boquet, Iván Pisa, Edith Cabrera and the rest of the people who have made my time at the UAB one full of experiences in my life path.

I cannot leave aside my friends from Medellin, who have always been there for me. I would like to thank Monica, Andrea, and Jennifer for their constant support, tenacity, and love I have received from them. To Pacho, Sebastian, Manuel and Santiago because their friendship has not changed over the years.

Finally, I want to thank all the love I have received every day from the people who are part of my home, you fill my heart with happiness. To my dear parents, my siblings Javier and Patricia for their constant support and valuable teachings. To my sister-in-law Viviana for her kindness and affection. To my nieces and nephews. I will never tire of thanking you for being the inspiration and engine of my days. Without your support, I would not have been able to complete this thesis.

Contents

Acronyms

AE	Autoencoder
AI	Artificial Intelligence
AKI	Acute kidney Injury
ANN	Artificial Neural Network
AUROC	Area Under the Receiver Operating Characteristic curve
CIFAR	Canadian Institute For Advanced Research
CKD	Chronic Kidney Disease
CNN	Convolutional Neural Network
CT	Computerized Tomography
CV	Cardiovascular
DA	Denoising Autoencoder
DL	Deep Learning
EBM	Evidence-Based Medicine
ECG	Electrocardiogram
EEG	Electroencephalogram
EHR	Electronic Health Record
EMG	Electromyography
ESRD	End-Stage Renal Disease
FN	False Negative
FP	False Positive

GAN	Generative Adversarial Network
GPU	Graphics Processing Unit
HD	Hemodialysis
HHTL	Hybrid Heterogeneous Transfer Learning
ICD-9	International Code of Diagnoses version-9
ICU	Intensive Care Unit
KDD	Knowledge Discovery in Database
KDIGO	Kidney Disease Improving Global Outcomes
LR	Learning Rate
LSTM	Long Short-term Memory
MAR	Missing At Random
MCAR	Missing Complete At Random
MI	Multiple Imputation
MIDA	Multiple Imputation Using Denoising Autoencoders
MIMIC-III	Medical Information Mart for Intensive Care-III
MNAR	Missing Not At Random
MRI	Magnetic Resonance Imaging
mSDA	Marginalized Stacked Denoising Autoencoder
MV	Missing Values
PCA	Principal Component Analysis
PPV	Positive Predictive Value
RF	Random Forest
RMSE	Root Mean Square Error
RNN	Recurrent Neural Network

ROC	Receiver Operating Characteristic curve
SDA	Stacked Denoising Autoencoder
SVM	Support Vector Machine
TL	Transfer Learning
TLAV	Transfer Learning by AVerage code
TLCO	Transfer Learning by COdes
TN	True Negative
TNR	True Negative Rate
TP	True Positive
TPR	True Positive Rate
WHO	World Health Organization

1 Introduction

1.1 Motivation

During the last decade, thanks to the arrival of Industry 4.0 (I4.0), artificial intelligence (AI) has evolved and gained traction in different fields that can transform our daily lives and society. One of the fields with the most significant impact during the adoption of such advances is healthcare [MS19]. It is estimated that the financial resources designated to the healthcare sector represent an average world gross domestic product (GDP) of 8%, with an annual increase of 4% during the last decade [Cha+19]. For the next two decades, investment is estimated to reach 16% of GDP worldwide. Thus, the growing interest in data captured in the healthcare sector, advances in communication technologies [APP18], and improvements in computing power have the potential of leading to a fascinating future where the generation of new knowledge and evidence-based on AI counteract the morbidity and mortality of the world's population.

The most general approach to generating knowledge and optimizing decision-making in the medical field is established on evidence-based medicine (EBM) [Sac97]. This paradigm integrates scientific evidence and the expertise of physicians for the prevention, diagnosis, and treatment of diseases. However, in clinical practice, many decisions made by health care professionals are not guided by the best evidence [MMC15] or are based on the experience of a few experts [Bur+18; Bur+19; Cov+18]. Moreover, the production and rapid availability of massive digitized data, the heterogeneity of patient information, and the inability of humans to process the vast amount of data as a whole have enabled the exploration of machine learning (ML) techniques that can reinforce EBM.

ML techniques automatically discover the knowledge needed to perform a learning task to support clinical decision-making. Deep Learning (DL)

techniques, as part of ML and inspired by the human brain, allow extracting complex relationships in data. Such representations have the potential to improve learning tasks in a medical setting. DL has demonstrated exceptional performances and potential in computer vision [Rus+15], natural language processing [SVL14] and speech recognition [Hin+12]. Therefore, DL can play an essential role in the paradigm shift from evidence-based medicine to data-based medicine.

The benefits of integrating DL solutions in a medical environment are reflected in the performance of the learning models and the solution to the challenges related to the medical pathology data itself. Such challenges include:

- Improving learning models.
- Addressing the absence of information.
- Managing the heterogeneity of medical records and the volume of information.

Addressing the first challenge could bring the inclusion of data-based models for clinical decision support closer. Using as much data as possible, even if it contains incomplete information, would reduce biases due to eliminating or substituting constant values in such records. Finally, integrating several data sources could open the way to learning tasks supported by multihospital data and improve the generalization of learning models by increasing the volume of information. Considering these challenges from a DL perspective, we will address the following research objectives.

1. Improve predictive models using data from pathologies with few patients and find the mechanisms to combine heterogeneous data based on DL.
2. Tackle issues related to the quality of clinical data, i.e., data sparsity, class imbalance, and the treatment of missing values (MV) through DL techniques.
3. Define mechanisms for transferring knowledge from massive data from other hospitals to support clinical hypotheses in units with small volumes of information based on DL.

The potential to address the issues mentioned above by DL has been the inspiration for this dissertation. This dissertation contributes to studying DL-based solutions to enhance learning and data-driven problem-solving tasks in a clinical data environment. This study begins by applying DL models for mortality prediction for patients in end-stage renal disease (ESRD), combining heterogeneous clinical data and exploring temporal dependencies. Then we address the replacement, the so-called imputation, of missing information, extending a multiple imputation methodology based on latent representations of data from DL. Finally, transfer learning (TL) approaches are explored to use massive data from large medical units in order to increase samples in medical units with a small volume of information. The transfer of knowledge aims to improve learning models for mortality prediction in ESRD.

1.2 Outline

This dissertation is structured as follows:

Chapter 2 presents relevant concepts that are necessary for this dissertation. It describes the data mining process and information related with clinical data. This chapter describes the most commonly used learning models in medicine and important performance metrics for this dissertation. Finally, the DL techniques used in this dissertation and the challenges to be addressed from a DL perspective are described.

Chapter 3 addresses the first objective of this dissertation. In this application, heterogeneous clinical data are combined and different groups of variables are explored together with their temporal dependencies to improve predictive models of mortality in ESRD patients. Different groups of variables are explored in this application, some suggested by the expert staff and others found automatically by ML mechanisms. However, the models generated with all variables provided better performance at different mortality windows. The models designed offered a better predictive capacity than those in the literature, providing evidence that such models can be used to support medical decisions that can influence the trajectories of a pathology.

Chapter 4 covers the second research objective. It is addressed the imputation of MVs, using a multiple imputation (MI) approach. Such an approach is based on an initial imputation of several copies of the data encoded into latent representations and combined into an average one, which we refer to as an average code. Such code is then decoded by the AE and replaces the missing information. This mechanism is validated with several benchmarks in the literature, presenting a better construction capacity than other popular imputation mechanisms.

Chapter 5 continues to address the challenges in clinical data. Thus, a TL approach is presented to increase the volume of data in mortality prediction models in ESRD patients. Such increase is carried out in two directions by increasing the number of samples and the feature space. In this solution, knowledge was transferred from data related to mortality in AKI patients, whose source is from first-level hospitals. Thus, the approaches present a considerable improvement in the learning models in ESRD and allow combating other subreddits such as sample imbalance since the presented approach gives the flexibility to carry out the transfer of only imbalanced samples.

Chapter 6 concludes this PhD's dissertation with a summary and discussion of the results obtained. Open lines of research and future work are also included in this section.

1.3 Research contributions

The main contributions presented in each chapter of this dissertation are detailed below.

1.3.1 Chapter 3

The main contributions of this chapter are related to the integration and structuring of several sources of information in the clinical setting to improve predictive models in ESRD using DL. The contributions have been published in a journal.

- E Macias, A Morell, J Serrano, JL Vicario, J Ibeas, ”Mortality prediction enhancement in end-stage renal disease: A machine learning approach”, in Informatics in Medicine Unlocked Journal, vol. 19, pp. 100351, 2020.

1.3.2 Chapter 4

This chapter contributes to the study of imputation mechanisms based on DL. The main results of this research has been published in two conferences and is under review in a journal.

- E. Macias, G. Boquet, J. Serrano, J. Vicario, J. Ibeas and A. Morel, ”Novel Imputing Method and Deep Learning Techniques for Early Prediction of Sepsis in Intensive Care Units,” 2019 Computing in Cardiology (CinC), 2019.
- E. Macias, J. Serrano, J. L. Vicario and A. Morell, ”Novel Imputation Method Using Average Code from Autoencoders in Clinical Data,” 2020 28th European Signal Processing Conference (EUSIPCO), 2021
- E. Macias, A. Morell, J. Serrano and J. Vicario, ”Multiple Imputation Using the Average Code from Autoencoders”, in Computer Methods and Programs in Biomedicine Update, vol. 2, pp. 100053, 2022.

1.3.3 Chapter 5

The contributions in this chapter are part of TL's novel mechanisms for the improvement of learning models in a DL-based clinical setting. The work in this chapter has been presented at a conference and is currently submitted to MDPI Electronics.

- E. Macias, J. Ibeas, J. Serrano, J. Vicario and A. Morell, ”Transfer learning and data augmentation for mortality predictive models in kidney disease”, 2020 28th European Signal Processing Conference (EUSIPCO), 2021.

- E. Macias, J. L. Vicario, J. Serrano, J. Ibeas and A. Morell, ”Transfer learning improving predictive mortality models for patients in end-stage renal disease”, in Electronics. Submitted, 2022.

1.3.4 Other contributions

During the PhD period, several collaborations have been carrried out, next they are listed:

- O. Gallés, N. Monill-Raya, E Macias, A. Morell, J. Serrano, D. Rexach, J. Comas and J. Ibeas, ”POS-382 DEEP LEARNING-BASED PREDICTION FOR MORTALITY IN PATIENTS WITH CHRONIC KIDNEY DISEASE: A NEW MODEL DEVELOPED WITH DATA FROM 10.000 PATIENTS OVER THE LAST 11 YEARS”, Kidney International Reports,. 7, 2022.
- J. Ibeas, N. Monill-Raya, E. Macias, C. Rubiella, J. Vallespin, J. Merino, E. Criado, J. Guitart, J. Lopez Vicario, A. Morell, J. Serrano, ”MO766 EARLY ARTERIOVENOUS FISTULA FAILURE PREDICTION WITH ARTIFICIAL INTELLIGENCE: A NEW APPROACH WITH CHALLENGING RESULTS”, Nephrology Dialysis Transplantation 36 (Supplement1), gfab103. 004, 2021.
- J. Ibeas, E. Lleal, E. Macias, C. Rubiella, A. Morell, J. Serrano, J. Lopez Vicario, ”SO019 A PREDICTIVE MODEL OF MORTALITY IN ACUTE RENAL FAILURE IN THE CRITICAL PATIENT: USEFULNESS OF ARTIFICIAL INTELLIGENCE”, Nephrology Dialysis Transplantation 35 (Supplement3), gfaa139. SO019, 2020.
- J. Ibeas, E. Macias, C. Rubiella, A. Morell, J. Serrano, A. Rodriguez-Jornet, J. Lopez Vicario, D. Rexachs, ”SP689 RENAL FAILURE AND MORTALITY: FROM EVIDENCE TO ARTIFICIAL INTELLIGENCE, CHANGE OF PARADIGM?”, Nephrology Dialysis Transplantation 34 (Supplement1), gfz103. SP689, 2019.
- E Macias, A Morell, J Serrano, JL Vicario, ”Knowledge extraction based on wavelets and dnn for classification of physiological signals: Arousals case”, 2018 Computing in Cardiology Conference (CinC) 45, p.p 1-4, 2018.

- E Macias, A Morell, J Serrano, JL Vicario, ”Extraction of knowledge of physiological signals based on deep neural networks”, 4th European Congress on Cardiology and eHealth, 2017.

2 Background

This chapter presents an overview of the methods used to extract and generate knowledge from medical data based on DL. We start by covering the main components of the knowledge discovery pipeline focusing on medicine. Later, some challenges related to medical data are discussed. Then, it is reviewed how AI has evolved from ML to DL. The main ML techniques used in a medical setting are covered, and finally, the DL methods used in this dissertation are presented at the end of the chapter.

2.1 Introduction

AI is a term that refers to the use of computers to mimic the cognitive behavior of humans to learn and solve tasks. The origin of this term was introduced by John McCarthy in 1956 [MB19]. However, Alan Turing demonstrated earlier, with his test, the possibility of machines to simulate human behavior. AI arose with the implementation of systems based on hard-coded rules designed by experts to solve specific tasks. Such systems were labor-intensive and time-consuming as expert personnel needed to maintain them and update rules manually. These types of expert systems were prevalent in the 1980s and 1990s. However, they presented issues dealing with unusual scenarios and heavily depended on expert personnel. In contrast to such systems, ML, as part of AI, appears as a field that gives computers the ability to learn without being explicitly programmed, i.e., learning based on examples. ML has come to play a leading role in the medical field because it can be integrated into multiple learning tasks. Among the most applied tasks are prognosis and diagnosis of diseases. In this way, it is possible to find patterns that allow determining the trajectories of a patient's health status, e.g., in the early prediction of cancer [Coc97; Kon01; Sun+07; EGF11; Mad+13] or to support survival analysis [Ish+08; Hsi+11; Coo+97; Mot+17].

Among the most relevant ML frameworks are artificial neural networks (ANN) and their evolution, DL. Their creation is inspired by the McCulloch Pitts neuron model, which represents the behavior of neural networks [MP43]. Although ANNs had a promising future in their beginnings, the technological limitations during the 90s did not allow them to demonstrate the scope they have reached today. For several decades they entered what is known as the winter of AI. This period was left behind thanks to the emergence of graphical processing units (GPU) that enhanced computing capacity, the improvement of storage and processing, and the vision of research groups such as the Canadian Institute For Advanced Research (CIFAR) [1] in not abandoning research on ANNs. As a result, ANNs have demonstrated their potential and continuous development to reach extremely complex models based on what is known as DL [GBC16].

In recent years DL has attracted attention in the medical field for its benefits in terms of volume of information to process, the ability to extract complex patterns and flexibility of heterogeneous data analysis [Chi+18]. Thus, different pieces of information collected in a medical environment can be analyzed individually or be integrated into a DL environment to extract complex patterns that would be difficult to address by expert staff or other ML approaches [Yue+20]. Thus, analogous to the evolution of AI towards DL, EBM is evolving from rule-based systems that are systematically reviewed and updated by expert personnel to systems supported by massive data that could soon support clinical decision-making. In the rest of this chapter, we will discuss the main components of an IA environment, focusing on DL powered by clinical data.

2.2 Electronic health records and knowledge discovery

Over the last decade, the digitization of patient health information has been massively adopted in most developed countries [Blu09; AK19]. It is estimated that around 96% of primary health care entities in European

[1] https://cifar.ca/

Union countries have adopted the collection of information in a digital format [@Com19]. The analysis and availability of such massive digitization have the potential to improve the quality, the efficiency of the healthcare service, and health outcomes of patients [AK19; Shu19]. Such digital records are called electronic health records (EHR) and are digitized information of patients' health follow-up, systematically collected in healthcare entities. EHRs store sequential information about demographics, laboratory results, diagnoses, medical history, medication, imaging, physiological signals, and narrative. They also track information regarding the management and billing of healthcare entities.

The information collected in EHRs is a combination of data gathered in clinical repositories whose information can be divided between those EHRs that have a tuple-wise structure and those that do not. Thus, for those records stored tuple-wise, each encounter with the healthcare entity generates a tuple where the patient's physiological variables are recorded. This information is stored in a table-like structure where its rows represent encounters and the columns the variables measured. The other EHRs differ in the format in which they are stored. These formats usually are stored as text or have time series structures such as physiological signals, e.g., ECGs, EEGs, or EMG; 2D or 3D medical images such as MRI, CT scans, or ultrasound. However, these types of EHRs can be transformed tuple-wise under characterization processes. In this dissertation, we are focused on the analysis of tuple-wise EHRs.

The analysis of EHRs can improve the understanding of pathologies and enhance patients' quality of life and healthcare services. All EHRs are potentially valuable to support the abovementioned benefits in an ML environment. However, to enable the integration of EHRs into ML or DL solutions, it is necessary to consider the knowledge discovery based in database (KDD) process [FPS96] commonly referred to as data mining. Thus, from an iterative process, as shown in Figure 2.1, it is possible to generate knowledge following the next stages:

- Data integration: this is the process where information from multiple databases is explored with the expert staff to ensure that the information collected is adequate to support the learning task. The data

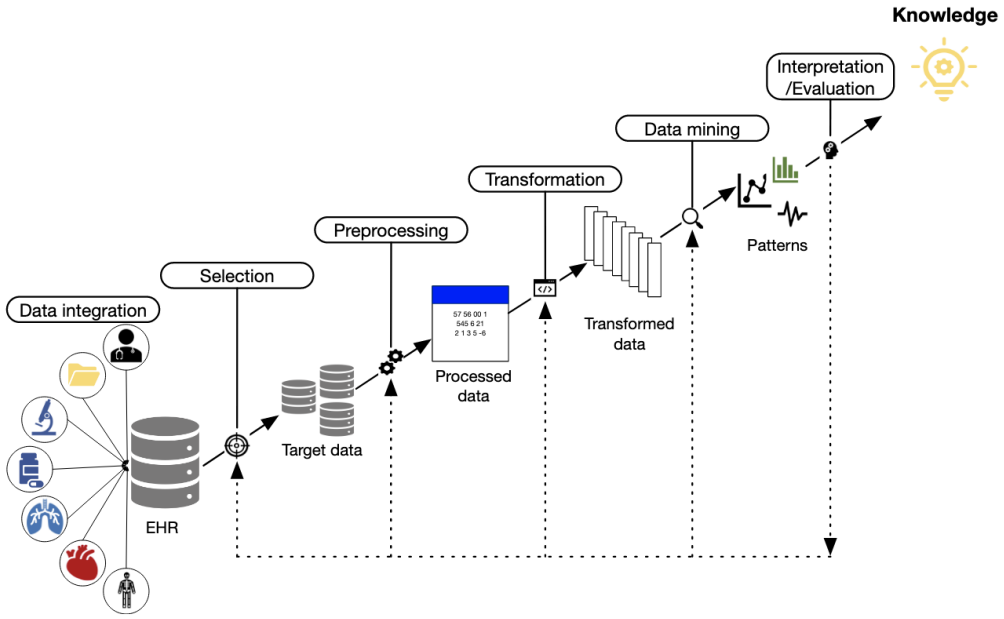


Fig. 2.1: Knowledge discovery in EHRs as a medical data value chain.

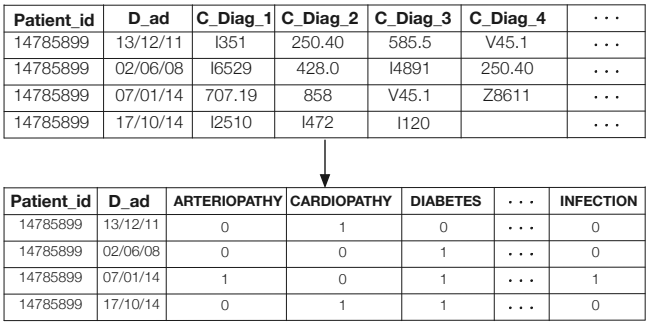
gathered are heterogeneous EHRs because they come from different sources, and their structure can be variable.

- Selection: it is defined as the process in which data relevant for a clinical study or clinical trial is taken from clinical databases. At this stage, it is of utmost importance to have expert knowledge in the field of study, to know the volume of data and the limitation that clinical data may have [SS14], to ensure that good quality data and cohorts support studies. In this stage, different mechanisms are followed to choose the right information to proceed with a study. Such mechanisms integrates best practices guidelines [Gof+14; Hip+08; DAg+08b; Rid+07] and protocols [Moo+15].

- Preprocessing: it is one of the most crucial stages of the KDD process because of the issues it tackles. Incomplete information, outliers, information redundancy, inconsistencies, imbalanced data, among others, are part of data preprocessing challenges [BIF18; GLD00]. Typical scenarios where the EHRs contain errors due to the omission of information by the patient, errors in the digitization of the information, or registration of erroneous values of medical equipment are addressed in this stage. It is estimated that 70-80% of the work in KDD is expended

at the preprocessing stage [Duh+03; Pér+15]. This stage ensures the quality and reliability of the data to support medical hypotheses. It aims to refine the characteristics chosen in the previous step and solve the drawbacks that the EHRs may have.

- Transformation: the data are consolidated into standard formats suitable for learning algorithms. This process is usually carried out through information mappings. Mapping elements from a source base to the target is performed to capture transformations. Figure 2.2 shows an example where diagnoses coded with the international code of diagnoses version 9 (ICD9) are transformed to a table of some diagnoses where the presence or not, 1 and 0, of a series of diagnoses in the encounters of a given patient are included. Other transformations that data can have are related to extracting valuable features from the same data. Dimensionality reduction eliminates attributes or by resorting to data representations in different spaces, e.g., using principal component analysis (PCA).



Patient_id	D_ad	C_Diag_1	C_Diag_2	C_Diag_3	C_Diag_4	···
14785899	13/12/11	I351	250.40	585.5	V45.1	···
14785899	02/06/08	I6529	428.0	I4891	250.40	···
14785899	07/01/14	707.19	858	V45.1	Z8611	···
14785899	17/10/14	I2510	I472	I120		···

Patient_id	D_ad	ARTERIOPATHY	CARDIOPATHY	DIABETES	···	INFECTION
14785899	13/12/11	0	1	0	···	0
14785899	02/06/08	0	0	1	···	0
14785899	07/01/14	1	0	1	···	1
14785899	17/10/14	0	1	1	···	0

Fig. 2.2: Transforming ICD9 codes to categorical attributes.

- Data mining: it is often used as a synonym for KDD, which is the process where intelligent mechanisms are applied to find practical and sometimes hidden patterns in data. Algorithms including association rules, classification, clustering, prediction, and pattern evaluation are usually used at this stage. For this dissertation, DL models are the focus of this stage.

- Interpretation: in this step, the patterns and rules are evaluated and interpreted through metrics specific to the data mining task. This step focuses on the understanding and usefulness of the model found in the

data mining stage. An iterative feedback process is used to enrich the previous steps from this stage.

The ultimate goal of KDD is the generation of knowledge. In this dissertation, the knowledge extraction mechanisms are based on DL techniques. Before we address them, it is necessary to understand the data and the main challenges at the preprocessing and transformation levels, to ensure that data quality is not a constraint for the models' performance.

2.2.1 Attributes in EHRs

Data are categorized into structured, semi-structured, and unstructured. The structured ones are those that follow a well-defined scheme. These schemes follow a table format where rows collect a set of attributes, organized in columns, of the data set. A dataset structure can be seen in Fig. 2.3. In a medical environment, the patient's encounters with the health entity are collected in each row. The columns refer to the attributes measured during such encounters. In this dissertation, attributes, characteristics, and features refer to the same concept. The second category refers to data with no predefined schema or identifiable structure. Examples of this type of EHRs include medical images, video, clinical notes, and audio. Semi-structured data is data that uses a self-describing schema such as XML or JSON files.

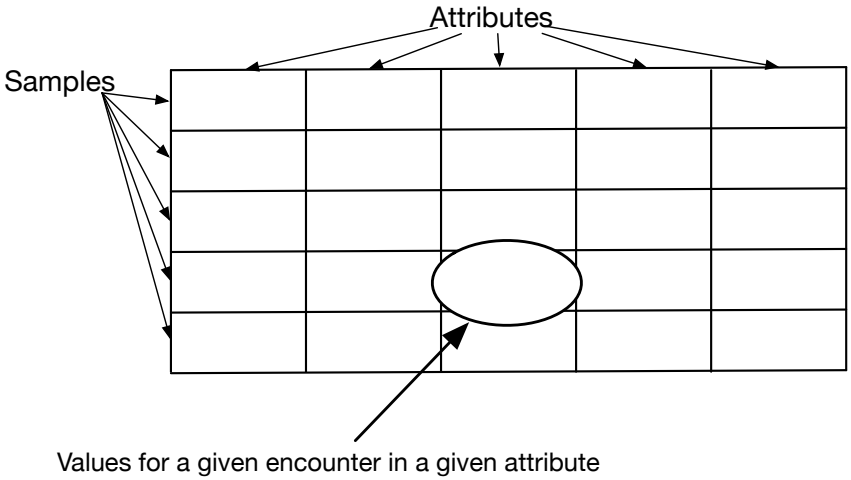


Fig. 2.3: Example of dataset with structure data.

In this dissertation, we will focus on structured data. Let us consider how to distinguish the attributes between dependent and independent ones. The values of dependent attributes are affected by other ones. Independent attributes do not release their values in other ones. This distinction is helpful in the data preprocessing stage as highly correlated attributes can be reduced or in the data mining stage as predictions of dependent variables can be made based on the independent ones.

On the other hand, they can be separated as numerical and categorical based on the attribute values. Numerical ones include real values with an order and a distance relation. In contrast, categorical attributes are those that have a symbolic representation. For instance, in a medical environment, the sex of a patient or the stage of chronic kidney disease (CKD) are some categories that represent information. In this type of representation, attributes take values in a finite range of possibilities. Categorical attributes are divided into nominal and ordinal attributes. If the attribute does not have a defined order, they are nominal. In contrast, ordinal attributes implicit a defined order. As an example, we can again take CKD. In this case, we can take the stage of the disease in a patient as an ordinal attribute because it represents the severity of the disease. The last way to distinguish the variables is their behavior concerning time. Those data that do not change over time are considered static data, while those that vary are called dynamic or temporal data. This type of data collects the evolution of patients' physiological attributes in a clinical setting.

2.2.2 Structuring and transforming EHRs

After knowing the structure and type of data used for the experiments, it is possible to apply different techniques to ensure that the learning models are fed with information in the correct format and to ensure that attributes are transformed to the same range of values so that their magnitude does not add bias to experiments. Thus, mechanisms such as data normalization and coding of categorical variables enable achieving the above.

Normalization

Normalizing the data involves scaling the features to a fixed range of values without loss of information. Scaling to a common range prevents variables with significant differences in their values from biasing learning models. Thus, there are several ways to normalize the data. Among them is to normalize according to the minimum and maximum values and according to the standard distribution, Eq. 2.1 and Eq. 2.2, respectively.

$$\mathbf{x_i}_{norm} = \frac{\mathbf{x_i} - \mu_i}{\sigma_i}, \tag{2.1}$$

where $\mathbf{x_i} \in \mathbb{R}^{n,1}$ is a feature of a dataset $\mathbf{X} \in \mathbb{R}^{n,f}$, with n samples and f features. μ_i and σ_i are the mean and the standard deviation of $\mathbf{x_i}$.

$$\mathbf{x_i}_{norm} = \frac{\mathbf{x_i} - min\left(\mathbf{x_i}\right)}{max\left(\mathbf{x_i}\right) - min\left(\mathbf{x_i}\right)}, \tag{2.2}$$

where $max\left(\mathbf{x_i}\right)$ and $min\left(\mathbf{x_i}\right)$ are the maximum and minimum value of $\mathbf{x_i}$.

Encoding categorical attributes

Categorical variables must be transformed into numerical values so that learning models can use them. This process is known as categorical encoding and has two main approaches. The first one, labeling encoding, encodes ordinal categories with string labels to integer values. The second category handles nominal variables so that each category generates a binary variable where 0 represents the absence and 1 the presence of the variable, this approach is also called one-hot encoding. Fig. 2.4 shows an illustration of such approaches.

2.2.3 Challenges related to data

To carry out any process of knowledge generation driven by data, it is necessary to know in detail the challenges we face in different aspects related to applications and the data itself. In a medical environment, there are

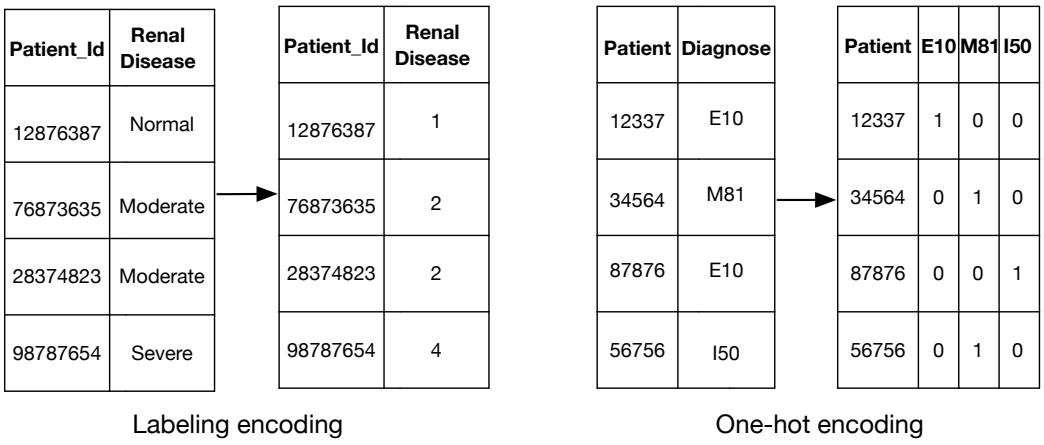


Fig. 2.4: Example of categorical coding in EHRs.

challenges associated with dealing with low-volume applications, integrating heterogeneous data sources, and missing data imputation to improve learning models.

Missing information

One of the most common issues in clinical data is missing information. This lack of information may be due to several factors, such as erroneous readings in medical measurement equipment, lack of data entry or errors in data collection by medical personnel, missing information due to the regularity in the collection of variables, integration of heterogeneous data in the EHR, among others. Fig. 2.5 illustrates the imputation pipeline.

The occurrence of MVs is inherent in a medical setting. Understanding their source of generation and structure allows us to use existing information to impute MVs. Let us consider the types of MVs. Its relationship with the data attributes can characterize them. MVs are classified based on three mechanisms [Rub76]: missing completely at random (MCAR), missing at random (MAR), and missing not at random (MNAR). MCAR refers to the fact that the appearance of the MVs does not depend on the data itself; missingness appears as a random process. Forgetting to include patient attributes is considered an MCAR since it does not rely on the patient or the missing attribute. MAR refers to the fact that the appearance of MVs depends

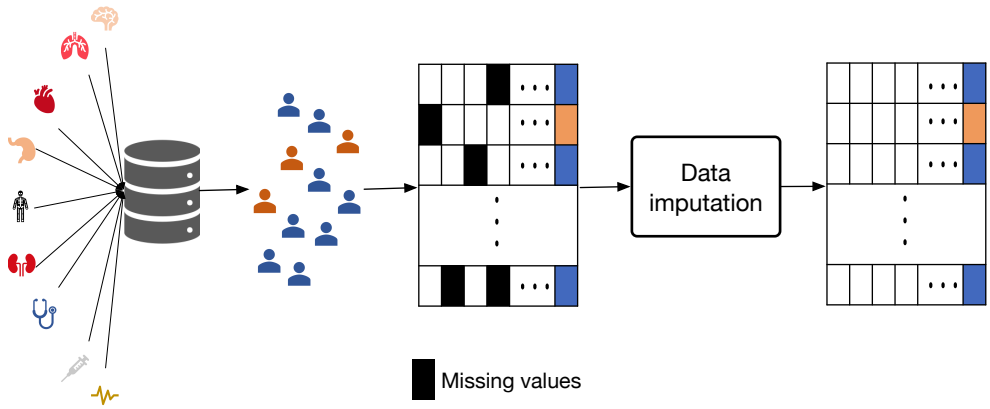


Fig. 2.5: Imputation process for EHRs.

partially on the observed data but not on the missing information. The abrupt stop in measuring temperature in a patient whose health condition worsens in an intensive care unit (ICU) is an example of this type of MVs. In this case, the appearance of MVs in temperature depends directly on a variable that indicates the patient's state in the ICU. Finally, MVs are classified as MNAR when directly related to their values. For instance, a depression survey is more likely not to be answered by patients with depression.

Volume of information

Although there is a growing movement towards the massive collection and integration of data from healthcare entities, most have a limited volume of information when studying pathologies individually. For instance, a nephrology unit of a tertiary hospital may have information on a cohort of fewer than 300 patients [Mac+20]. Such a challenge in volume opens up the possibility of exploring mechanisms for knowledge transfer among healthcare entities to support those that do not have a significant volume of data. Thus, it is necessary to take advantage of diverse sources of information to strengthen learning models in pathologies with few data. Therefore, several mechanisms allow the integration of other data sources through a paradigm known as TL. Fig. 2.6 shows an illustration of a TL process.

TL uses the knowledge gained from one domain in another. The domain from which such knowledge is extracted is known as the source domain and

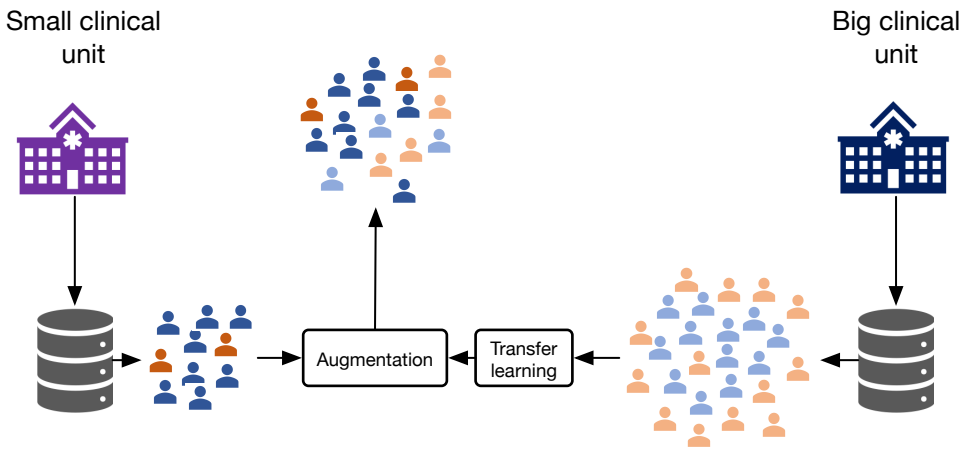


Fig. 2.6: Transfer learning process in healthcare.

the one that uses it to reinforce learning tasks is known as the target domain. TL has been used in other fields to support image and text classification tasks. However, in the medical field, it is still in its infancy.

In the field of TL *what to transfer*, *how to transfer* and *when to transfer* are the research questions in this field. What to transfer refers to the part of the decided knowledge that could be transferred between the domains. Once the first question is covered, the second question is addressed to define the algorithms' mechanisms to transfer knowledge. Finally, when to transfer will be based on the transferred knowledge's contribution to a learning task. If this contribution outperforms the learning task, it is known as positive TL, in contrast to having negative TL.

On the other hand, TL is categorized as inductive, transductive, and unsupervised. For the first category, the target task is different for both domains, whether the data come from the same domain. In contrast, the task is the same in transductive TL, but the data differ. The target task is different for unsupervised TL but related to source one. This type of TL is used to solve unsupervised learning tasks such as clustering or feature reduction in the target domain. This dissertation will focus on inductive TL to tackle the challenge of sample volume and sample imbalance in a source domain, supported by data derivated from kidney diseases. This topic will be further discussed in Chapter 5.

2.3 Learning from data

As Aristotle first stated in his *Treatise on Anima*, we are a tabula rasa; we learn according to our experience. Similarly, so do ML models. Tom Mitchell defined ML formally as follows: *A computer program is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E*. Thus, ML learns from experience and improves its performances as it learns. Learning is a process of looking for knowledge through experience.

ML's advances in medicine are related to analyzing data and discovering new patterns. The spectrum of applications in which ML techniques are used is broad. These include disease identification and classification[Fer+19; Dil+19], drug discovery [Kom+18], disease trajectory prediction [Har+19], medical imaging, personalized medicine [Frö+18; HJ14], among others [He+19]. Let us now consider the main components of ML and their evolution up to the integration of DL models.

2.3.1 Learning tasks

Learning tasks are those hypotheses that are supported by data. In the medical field, the design of these tasks is refined by expert knowledge and the definition of their objectives is not usually a trivial task. For example, let's consider a cohort of 1500 patients from a sleep unit of a tertiary hospital. Each patient in their sleep study has at least 7 hours of recording of different physiological signals. From these data, a diverse medical hypotheses can be generated to learn tasks such as detecting arousals, classifying sleep phases, and determining positive pressure to treat sleep apnea.

The learning process in ML can be classified as supervised, unsupervised, semi-supervised, and reinforcement learning. In supervised learning, models contain a target or label to which a set of input attributes are adapted through a mathematical mapping between input and output called a model. In contrast, unsupervised learning does not have a defined target. Instead, it looks for patterns within attributes that allow projecting this information to lower-dimensional spaces or finding clusters. Semi-supervised learning

combines labeled data with unlabeled data to find common hidden structures among these data sets to improve a learning task. Finally, reinforcement learning models are based on teaching an agent to choose an action that allows it to maximize rewards. In this dissertation, we will focus on supervised and unsupervised models.

In the supervised learning paradigm, two main tasks stand out, classification and regression. In classification a learning model is expected to find the category to which a set of data $\mathbf{X} \in \mathbb{R}^{s,f}$, with s samples and f features, belong to. To solve this task, the learning model generates a function $f(\cdot)$ that maps examples $\mathbf{x}^{(\mathbf{i})} \in \mathbb{R}^{1,f}$ to a class $\mathbf{y}^{(\mathbf{i})}$, i.e., $\mathbf{y}^{(\mathbf{i})} = f\left(\mathbf{x}^{(\mathbf{i})}, \boldsymbol{\theta}\right)$, where $\boldsymbol{\theta}$ contains the parameters of the model. For regression, the learning task is similar, with the difference that the output is not a category but a numerical value. Fig. 2.7 presents a graphical representation of three of the mentioned task.

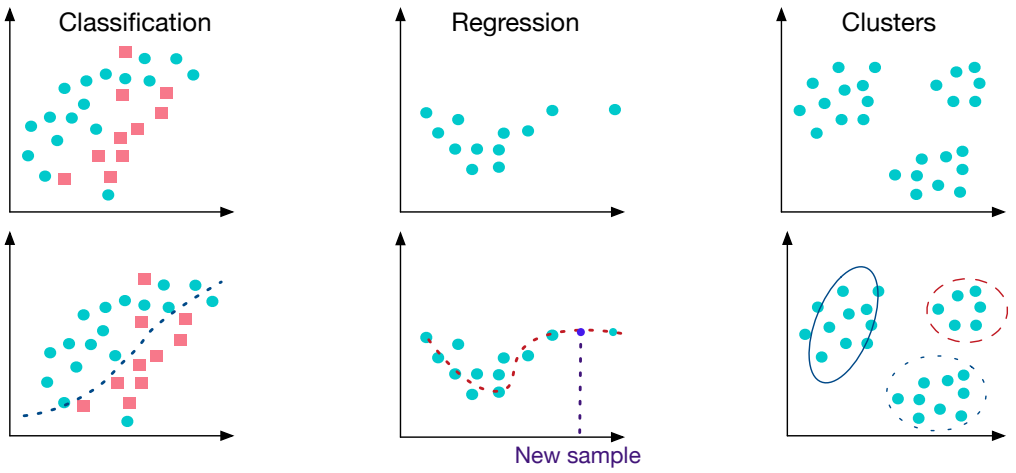


Fig. 2.7: Graphical representation of learning tasks.

2.3.2 Learning capacity: overfitting vs underfitting

The learning capacity of an ML model is usually evaluated through its complexity and factors such as underfitting and overfitting. Thus, before generating an ML model, the dataset is usually separated for training and test data. The first set is used to generate a learning model validated on new data. This validation is the initial concern of the learning models since the goal is to

get similar performance in test data as in the training ones. This process is known as model generalization and is defined as the ability of a model to perform correctly on new data. Thus, a model $f(\mathbf{X}, \boldsymbol{\theta})$ is trained, which objective is to find the proper parameters $\boldsymbol{\theta}$ that reduce the error between its predictions and the real target. The quality of the approximations is carried out through the computation of a loss function $L(\mathbf{y}, f(\mathbf{X}, \boldsymbol{\theta}))$. L measures the difference between the actual value and the output of the learning model in the presence of an input.

On the other hand, ML models are often exposed to underfitting and overfitting issues. Underfitting produces a low performance in training and test sets. The learning models are very simple compared to the complexity of the data and the learning task. In contrast, overfitting involves a complex model that fits the training data well but fails to generalize the test data. Such issues are often tackled using different approaches. One is to use learning models whose parameters have sufficient capacity to fit the data. For example, a linear model would not be adequate to deal with the data in Fig. 2.8, it would present underfitting problems. For overfitting, it is possible to increase the data sample to improve the generalization of the learning models. In the clinical setting, this is not commonly achievable as the data volumes of medical units are usually small.

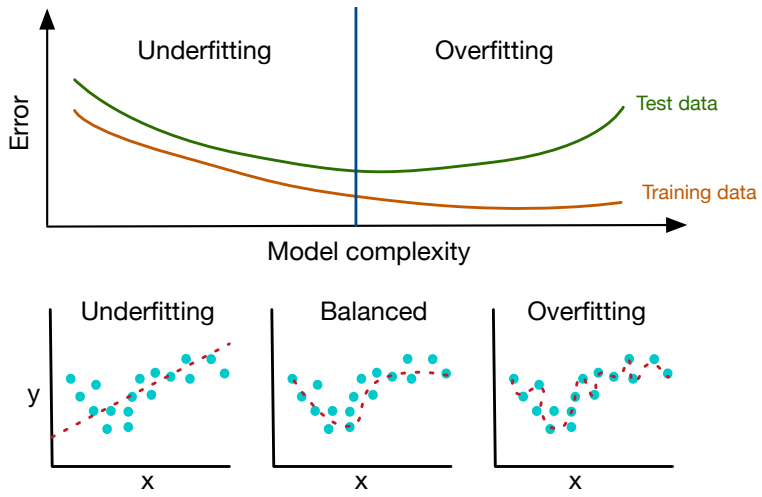


Fig. 2.8: Underfitting and overfitting issuees for ML models.

Another way to counteract overfitting problems is by penalizing the parameters of the learning model by adding a term to the loss function. L1 and L2

regularizations penalize these parameters, although they are used also for linear model, they will be covered in detail in the section dedicated to DL.

Finally, to better measure the performance of learning models, approaches such as k-fold cross-validation can be applied. The data are divided into k groups, each of them contains its training and test data. Then k models are trained with each group, and their average performance will be that of the model. This approach can be seen in Fig. 2.9.

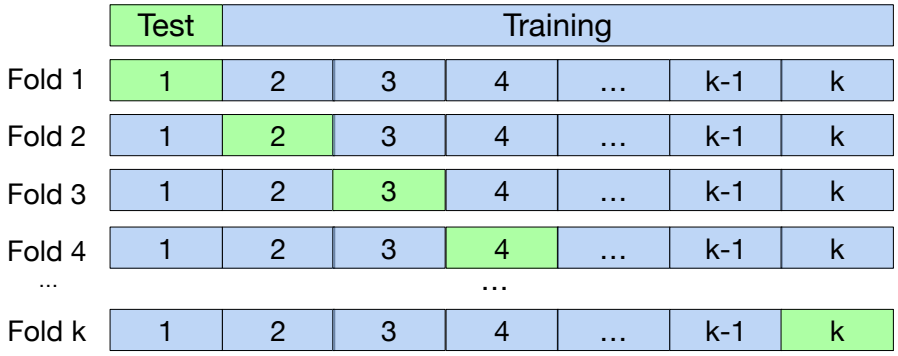


Fig. 2.9: K-folds cross-validation.

2.3.3 Metrics to measure performance in learning models

In order to choose an appropriate learning model, it is necessary to define metrics that allow us to compare the different options we have to approach our learning task. Thus, the evaluation of such models is carried out through a performance measure that will depend on the type of learning we wish to address.

Classification

For classification purposes, we usually use a confusion matrix (see Table. 2.1) that allows us to separate the classified samples into four groups:

- True positives (TP): samples that are classified correctly in the positive class;

- False positives (FP): samples that are classified as positive samples, but the real class is negative;
- False negatives (FN): samples classified as negative but with a real target in the positive class;
- True negatives (TN): are the negative samples that are classified correctly to the same outcome.

Tab. 2.1: Confusion matrix.

	True condition		
Prediction	Positive	Negative	Total
Positive	TP	FP	T_+
Negative	FN	TN	T_-
Total	D_+	D_-	

The performance measures that can be computed from this matrix are accuracy,

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{2.3}$$

which measures the overall performance of the classifier; the precision or positive predicted value (PPR),

$$PPR = \frac{TP}{TP + FP} = \frac{TP}{T_+} \tag{2.4}$$

which indicates the percentage of correct positive samples that the classifier labelled as positive; sensitivity, recall or true positive rate (TPR),

$$TPR = \frac{TP}{TP + FN} = \frac{TP}{D_+} \tag{2.5}$$

which indicates the percentage of positive samples that were correctly labeled by the classifier; the specificity or true negative value (TNV)

$$TNV = \frac{TN}{FP + TN} = \frac{TN}{D_-} \tag{2.6}$$

which indicates the percentage of samples labeled as negative and belonging to that class; $F1$ score,

$$F1 = \frac{PPV \cdot TPR}{PPV + TPR} \tag{2.7}$$

which is a combination of precision and recall.

Another metric that usually provides relevant information on the performance of classification models is the receiver operation characteristic (ROC). ROC is the graphical representation of the behavior of the sensitivity and specificity of a classifier as a function of a decision threshold. The metric associated with the ROC is the area under the curve (AUROC), whose values vary from 0 to 1, with 1 being a perfect classification and 0.5 the performance of a random classifier. Fig. 2.10 shows what a ROC looks like with three different models. It can be seen that model A is better than B and C. In a clinical setting, the expert must decide which threshold value is appropriate, relying on the learning task so that he/she can give more importance to the specificity in the detection of pathology than to its sensitivity.

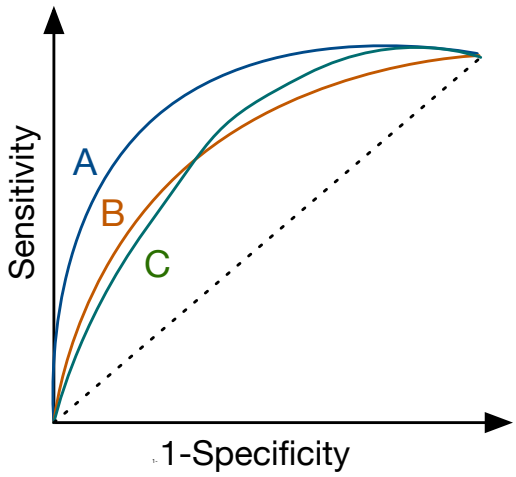


Fig. 2.10: ROC curve.

Regresion

When it is required to measure the performance of a model in the prediction of quantitative values, the root means squared error (RMSE) is usually used.

The RMSE (Eq. 2.8) measures the error between the prediction of a set of data as a function of its prediction.

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N} \|y_i - y_i'\|^2} \tag{2.8}$$

with a dataset of N samples, y_i the actual value and y_i' the predicted one.

2.3.4 Data driven methods in medicine

In recent years, the methods most widely used by the research community are linked to the performance they offer for the data and the benefits to address their inherent problems. Thus, several surveys reflect on the performance of models that can use the non-linear relationships that may exist in the data [FP+17; Jia+17; WLR19; Dar+20]. Such studies highlight that the most popular techniques within the scientific community are support vector machine (SVM) and ANN, followed by logistic regressions and random forest (RF). The following is a brief description of the mentioned ML techniques.

SVM

SVM is a method commonly used in classification problems that is based on defining a separable space that allows to distinguish classes based on the computation of hyperplanes and maximization of their margin or distance between hyperplanes [CV95]. The learning process starts with a dataset with its labels,

$$(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N), \quad y_i \in \{-1, 1\} \tag{2.9}$$

and the optimal hyperplane that distinguish classes,

$$\mathbf{w} \cdot \mathbf{x} - b = 0. \tag{2.10}$$

Fig. 2.11 shows the two dimensional representation of this hyperplane with other two that represent the margin,

$$\mathbf{w} \cdot \mathbf{x} - b = -1, \quad \mathbf{w} \cdot \mathbf{x} - b = 1. \tag{2.11}$$

The idea behind SVM is to maximize the the margin, the distance between both auxiliary hyperplanes, so that the regions to which the samples belong are as separate as possible and that their respective space is very well defined to classify the samples. Then, from any of the expressions in Eq. 2.12 can be found that based the distance can be expressed as $d = \frac{1}{\|\mathbf{w}\|}$. Finally, $\mathbf{w}$ is minimize to maximize the distance, with the constrain,

$$y_i \left(\mathbf{w} \cdot \mathbf{x_i} - b\right) \geq 1, \quad \forall i = 1, \ldots, N. \tag{2.12}$$

These two types of constraints (for $y_i = 1$ and $y_i = -1$) are to ensure that each class of points lies on one side of the dotted lines. The points that are right on the border are called supporting vectors and hence the name SVM.

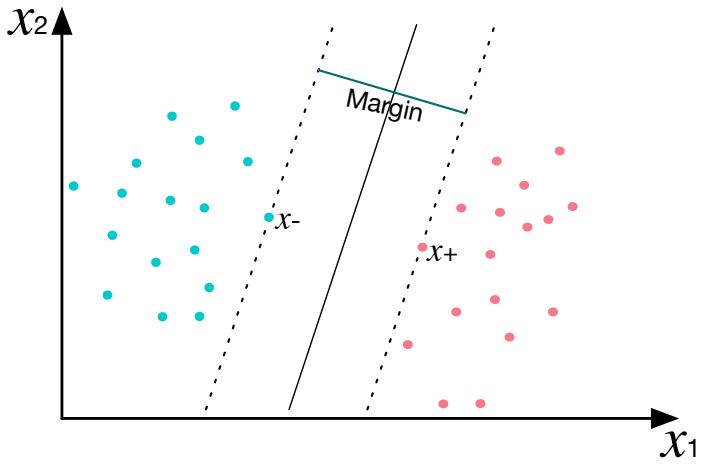


Fig. 2.11: SVM for two features.

The optimization problem for SVM uses Lagrange multipliers to find the parameters $\mathbf{w}$ and b,

$$L\left(\mathbf{w}, b, \alpha\right) = \frac{1}{2}\mathbf{w}^T \cdot \mathbf{w} - \sum_{i=1}^{N} \alpha_i \left(y_i \left(\mathbf{w} \cdot \mathbf{x_i} - b\right) - 1\right), \tag{2.13}$$

$$\frac{\partial L}{\partial \mathbf{w}} = 0 \rightarrow \mathbf{w} = \sum_{i=1}^{N} \alpha_i y_i \mathbf{x}_i \tag{2.14}$$

$$\frac{\partial L}{\partial b} = 0 \rightarrow \sum_{i=1}^{N} \alpha_i y_i = 0 \tag{2.15}$$

where α_i are the lagragians multipliers.

The inequality constraint of Eq. 2.12 is transformed to equality and then verified to meet the karush-Kuhn-Tucker (KKT) conditions [GT12], in order to find the parameters in the optimization, i.e.,

$$\alpha_i \geq 0, \tag{2.16}$$

$$\alpha_i \left[y_i \left(\mathbf{w} \cdot \mathbf{x_i} + b \right) - 1 \right] = 0. \tag{2.17}$$

Then, by replacing Eq. 2.14 and Eq. 2.15 in Eq. 2.13, it is possible to find the dual problem [CV95] that allows maximizing the Lagrangian subject to the dual variables being positive, that is normally solve using sequential minimal optimization [Pla98].

$$L_D = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x_i} \mathbf{x_j}. \tag{2.18}$$

Then, b is found based on the values for α_i in Eq. 2.17 solving the equation

$$\left(\sum_{i=1}^{N} \alpha_i y_i \mathbf{x_i} \cdot \mathbf{x} \right) + b = 0. \tag{2.19}$$

Finally, the samples are classified using a sign function that separate the classes, i.e.,

$$f\left(\mathbf{z}\right) = sign\left(\mathbf{w} \cdot \mathbf{z} + b\right) = sign\left(\sum_{i=1}^{N} \alpha_i y_i \mathbf{x_i} \cdot \mathbf{z} + b \right). \tag{2.20}$$

Decision trees

Decision trees are diagrams of logical constructs based on data. They are based on consecutive rules that allow separating a set of data $\mathbf{X}$ to its respective target $\mathbf{y}$ with the minimum error. Thus, taking as an example the decision tree in Fig. 2.12, it has its origin in the root (top of the tree). Such node is the parent of the subsequent nodes, the so-called children nodes, where based on a condition, it is decided if the node ends up in a class (decision node) or continues propagating to the tree's leaves. The right side of Fig. 2.12 shows how the zones are separated according to the decision threshold in the decision nodes.

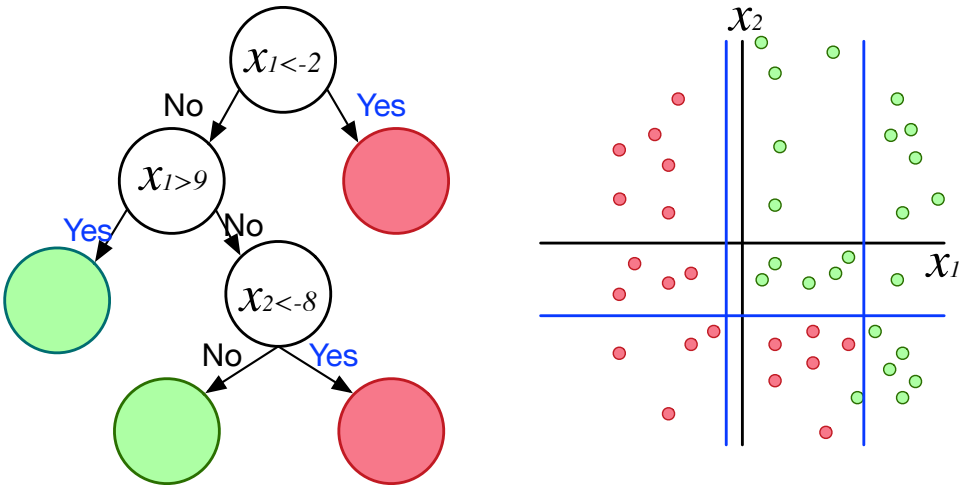


Fig. 2.12: Decision tree.

The learning mechanism of decision trees is based on the computation of the entropy of the parent nodes and the information gain [Qui14] contributed by the child nodes. Thus, the process starts by calculating the total entropy of the data on the target dataset,

$$E\left(\mathbf{y}\right) = \sum_{i=1}^{C} -p_i log_2 p_i, \tag{2.21}$$

where p_i is the probability of a sample of a class i in the dataset. Then, the entropy of children nodes are computed for each attribute as a function of the target, i.e, $E\left(\mathbf{y}|\mathbf{x_i}\right)$. Once such entropies are computed, the information gain (IG) is computed for each attribute as

$$IG\left(\mathbf{y}, \mathbf{x_i}\right) = E\left(\mathbf{y}\right) - E\left(\mathbf{y}|\mathbf{x_i}\right). \tag{2.22}$$

The feature that makes the IG smaller will be the one that decreases the uncertainty in the system and will serve as the new parent node for the following child nodes of the tree. This process is repeated for each feature of the dataset. Those nodes with the highest information gain will be the new parent nodes. The tree continues branching until each node reaches a zero entropy value or a fixed one to become a decision node.

Random forest (RF)

RF is an ensemble method, which refers to models that work together to solve a learning task. Thus, RF is a collection of decision trees working together. On top of decision trees, RF offers several features. The first one is called bagging and consists of using parallel learning models to solve a common learning task, see Fig. 3.3. For RF, this mechanism is executed using N decision trees with N datasets previously randomly separated. Thus, under a consensus mechanism, the outcome of the learning task is decided by a majority vote. The bagging mechanism reduces the overfitting problem caused by training a single decision tree that learns the training data, while a more robust model is achieved with bagging. However, some of the attributes of these subsets may be overemphasized, and the trees may be highly correlated if they share the same features. To avoid this issue, the samples and the features of the dataset are separated in subspaces for each of the decision trees. Thus, it is possible to restrict the decision trees to randomly use a maximum number of attributes for the learning task.

Another important feature offered by RF is that the importance of attributes can be found. This is achieved under different mechanisms that measure the performance contribution of the characteristics within each tree that make up the RF.Among the most used mechanisms are the measure of impurity of the nodes by entropy or by finding the importance of the attributes from the computation of the GINI index [NKW18].The importance of variables is measured according to how much they have served to reduce the impurity in the division of the tree where they have been used. Then it is necessary to

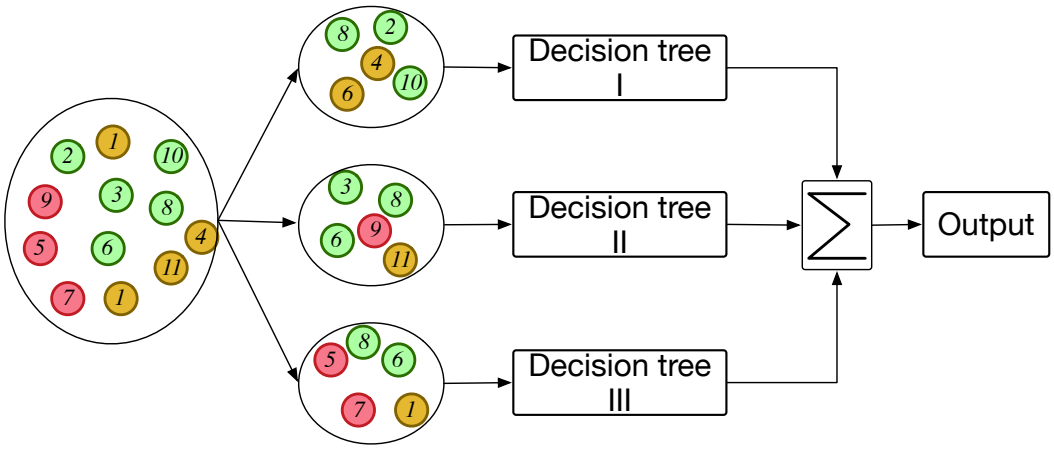


Fig. 2.13: Random forest using bagging mechanism to decide the outcome in a learning task.

average by weighting by the samples affected in the divisions involved and over how many divisions the variable is involved throughout all the trees of the RF.

2.3.5 DL models

DL models refer to specialized architectures based on ANN. The concept of deep is given by the number of hidden layers an ANN has, it is considered deep with more than two hidden layers. The ANN architecture is usually chosen based on the learning task to be addressed. Although most applications are based on ANNs with simple architectures in the medical field, other architectures may generalize better if there is a considerable amount of data. For temporal dependencies, recurrent neural networks (RNN) [RHW86a] are able to exploit temporal relationships in the data. AEs [RHW85] specializes in unsupervised learning and explores complex data representations in latent spaces. Other architectures out of this dissertation's scope are convolutional neural networks (CNN) [LB+95] that take advantage of spatial relationships, generative adversarial neural networks (GAN) [Goo+14] are used to generate synthetic samples based on adversarial players, among others. Before going in-depth into the study of each architecture, let us consider how ANNs work and the advantage they bring over other learning methods.

ANN

ANNs make it possible to exploit non-linear and often very complex relationships that are difficult to solve by other learning models. To see this effect, let us take a practical example of the effectiveness of a drug dose on the health condition of a set of patients. Thus, in Figure 2.14, it can be seen that at small and high doses of the drug, it does not affect the patient's condition, while at a medium dose, it will affect it. If we wanted to model the efficacy behavior as a function of dose with linear models such as linear regression, we would not find a straight line that would allow us to model the entire data set. In contrast, ANNs can fit these relationships and generate a function like the one on the right side of the figure and complete the learning task.

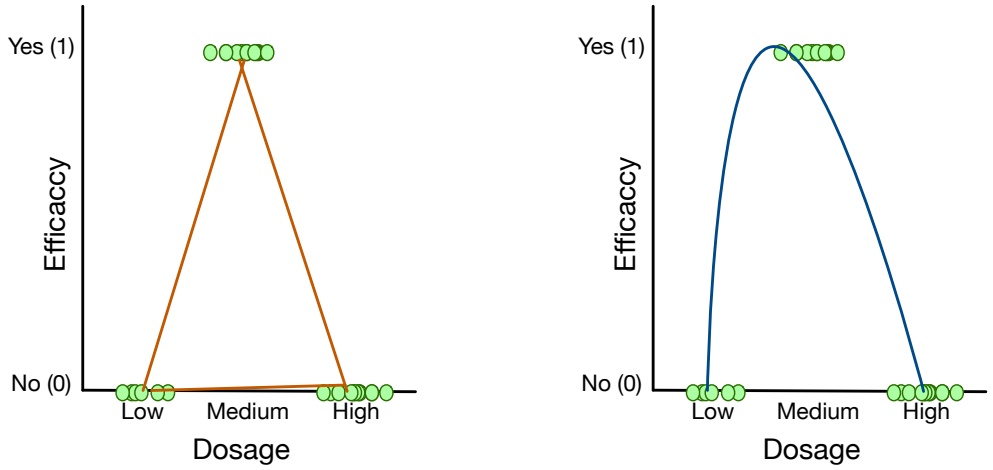


Fig. 2.14: Linear models behavior vs ANN behavior.

Structure: an ANN is composed of $L+1$ layers ($l = 0, \ldots, L$), where $l = 0$ represents the input layer, N_l neurons per layer, and the interconnections between neurons of consecutive layers are called weights. The graphical representation of an ANN can be seen in Fig. 2.15.

The goal of ANN is, given a set of N input examples $\mathbf{x^{(n)}}$, where $n = 1, ..., N$ and its corresponding classification targets $\mathbf{y} = [y_{1,n}, \ldots, y_{c,n}]$, to learn the best non-linear model that maps the input to its respective target. Thus, driven by a considerable number of training samples, an ANN can learn an optimized non-linear function in an iterative process to minimize the input and output error. Samples are first presented through the input layer,

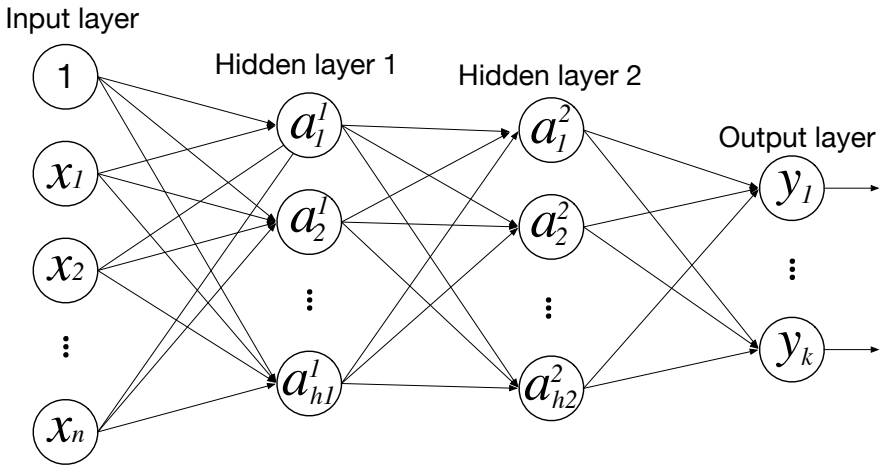


Fig. 2.15: Structure of ANN.

whose neurons connect with one or more hidden layers, and they link to the output layer where the result of the model is obtained. The output of the i-th neuron at the l-th layer (the so-called activation) is the linear combination of the outputs at the previous layer, taking into account the weights of such connections, and modified by a specific non-linear function $f(\cdot)$, usually the sigmoid, the hyperbolic tangent or the Rectified Linear Unit (ReLU). In other words,

$$a_j^l = f\left(\sum_{i=0}^{N_{l-1}} w_{i,j}^{l-1} a_i^{l-1}\right) \tag{2.23}$$

where $w_{i,j}^{l-1}$ is the weight that connects the i-th activation at layer $l-1$ to the input of the j-th neuron at layer l. Note that $a_0^l = 1$ in all layers except the output layer to consider the bias term. Some activation functions are presented in Fig. 2.16. Because these functions are non-linear, interaction throughout the ANN allows complex relationships within the data to be explored.

At the output layer, activations are usually normalized (e.g., softmax) so that the resulting values take a value between 0 and 1. They can be interpreted as a probability estimation; for example, y_k represents the probability that the input example belongs to the k-th class.

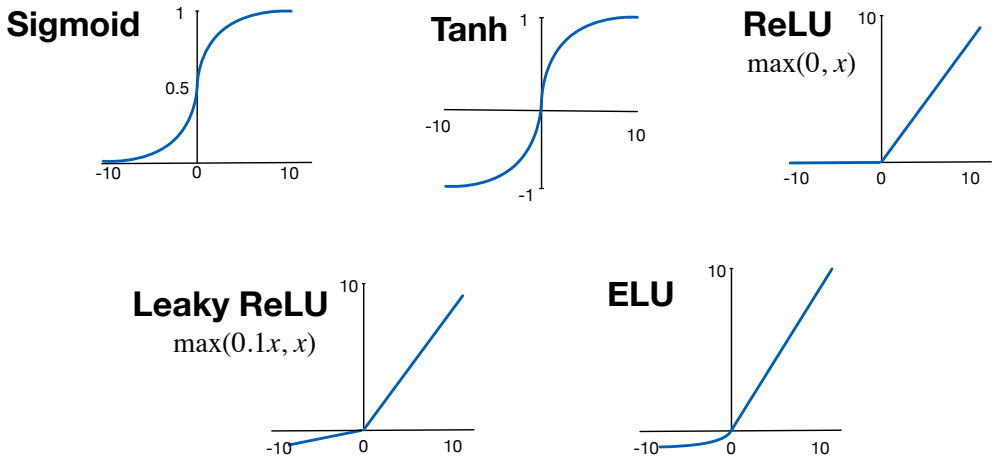


Fig. 2.16: Activation functions.

Training: this process is carried out in an iterative process where the network weights are adjusted to minimize a cost function that measures the error between predictions and the corresponding true values. As a starting point, let's consider cross-entropy as the loss function,

$$L = -\frac{1}{N}\sum_{\mathbf{x}}\sum_{k}\left[y_k log\left(a_k^L\right) + (1-y_k)\, log\left(1-a_k^L\right)\right] \tag{2.24}$$

where N is the total of samples, $\mathbf{y_k}$ is the k_{th} outcome (ground truth) and a_k^L the predictive value from the ANN at the output layer L.

The learning process in ANN is based on gradient descent [RHW86b] and back-propagation [Wer94]. Thus, from variations in the ANN's weights, the cost function is aimed to move towards a minimum, as shown in Fig. 2.17.

Thus, this process starts by initializing the ANN weights randomly. Then, the inputs are propagated along with the ANN until they reach its output. At the output layer, the loss function is computed, and from this point, the back-propagation algorithm starts. Algorithm. 1 shows how the weights of the ANN are updated based on the propagation of the gradient of the loss function. The process starts with the calculation of the gradients for all the activations $\mathbf{a}^{(l)}$ for each layer l, starting from the output layer to the first hidden layer. In this process the network parameters vary according to the gradients in order to reduce the error.

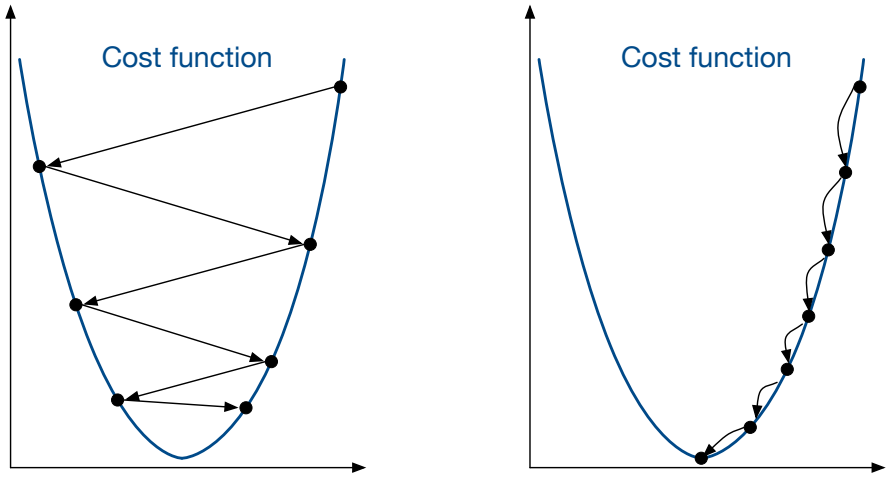


Fig. 2.17: Gradient descent.

This process is repeated until a minimum, not necessarily absolute, is achieved in the loss function. Thus, in every iteration, the parameters of the network are updated following the expression

$$\Theta' = \Theta - \alpha \nabla_{\Theta} L, \tag{2.25}$$

where Θ contains all the parameters of the ANN and α is the hyperparameter called learning rate and represents how fast the cost function will move towards a minimum. If this parameter is very large, the learning model will not reach a minimum, but if it is very small, it may take a long time to reach some minimum in the loss function. To combat these effects, techniques are often applied where this parameter is adapted during training. Some algorithms accelerate the learning process by dynamically changing α. Adaptive moment estimation (ADAM) [KB14] is one of the best performing approaches for this task.

On the other hand, it is common to have problems of overfitting, which happens when the network does not learn a model from the underlying data but memorizes the individual examples. To solve them, regularization mechanisms that penalize ANN weights or prevent the network from memorizing training data are commonly used. A common way to reduce this effect is applying L2 weight regularization [KH92], a quadratic penalty function is added to the weight, i.e. (2.24) is modified to

Algorithm 1 Back-propagation algorithm

Input: $\mathbf{y}'$, $L(\mathbf{y}', \mathbf{y})$

1 After forward propagation, compute the gradient on the output layer:

$\mathbf{g} \leftarrow \nabla_{\mathbf{y}'} L(\mathbf{y}', \mathbf{y})$

2 **for** l =l, l-1, . . . , 1 **do:**

Compute the gradient on the layer's output into a gradient into the pre-nonlinearity activation:

$\mathbf{g} \leftarrow \nabla_{\mathbf{a}^{(l)}} L = \mathbf{g} \odot f'\left(\mathbf{a}^{(l)}\right)$

Compute the gradient on weights and bias
$\nabla_{\mathbf{b}^{(l)}} L = \mathbf{g}$
$\nabla_{\mathbf{W}^{(l)}} L = \mathbf{g}\mathbf{h}^{(l-1)\top}$

Propagate the gradients w.r.t. the next lower-level hidden layer's activations:
$\mathbf{g} \leftarrow \nabla_{\mathbf{h}^{l-1}} L = \mathbf{W}^{(l)\top}\mathbf{g}$

$$L' = L + \frac{\lambda}{2N} \sum_{l=1}^{L} \sum_{i=1}^{N_{l-1}} \sum_{j}^{N_l} \left(w_{i,j}^{l}\right)^2 \tag{2.26}$$

With L2 regularization, controlled by λ, we limit the adaptation capacity of the network by penalizing large weights.

RNN

RNNs are a family of ANNs specializing in processing data with temporal sequences. In this type of architecture, the processed data considers the history of data that the network has already seen. This is achieved by adding internal memories in the network structure where the information is stored and used in future iterations in the learning process. Although RNNs offer better performance on sequential data, they are usually limited to short-term time dependencies. The drawback above has to be considered in a clinical environment because many pathologies are chronic and depend on past events. Moreover, classical RNNs present issues related to the explosion and vanishing of the gradient that updates the weights. In such a process, the

weights are not updated or grow too much. Thus, RNNs evolved towards Long short-term memory (LSTM) to solve such problems.

LSTM: are networks that were designed to solve the drawbacks of RNNs. The temporal dependencies in LSTMs are controlled by mechanisms that emphasize the sequence information flowing through the network. Thus, the central components of LSTMs are the cell state c_t, the hidden or previous state h_t, and its gates. The structure of the network can be seen in Fig. 2.18. It shows three types of gates, the forget gate f_t, the input gate i_t, and the output gate o_t. The cell state is transporting the information considered relevant by the gates.

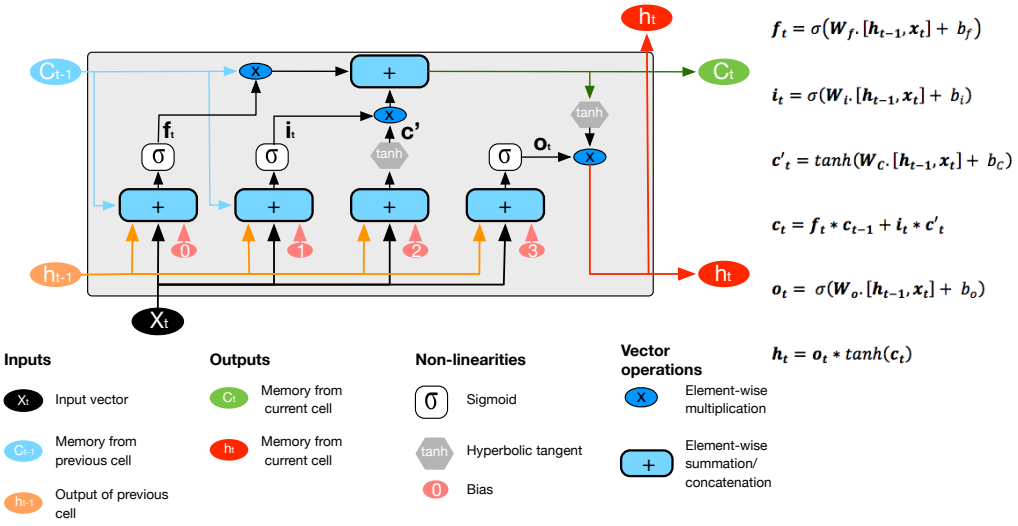


Fig. 2.18: LSTM cell, σ is the sigmoid activation function.

The cell gates are neural networks with fixed activation functions responsible for processing sequential information. Based on their activation function, they can distinguish the relevance of new data entering the cell. The core idea is to combine the information from the gates in c_t. The forget gate indicates which information from the combination of the previous state, h_t, and the input, x_t, is discarded based on the sigmoid function. If the value is near 0, then it is not relevant for c_t. Then, new information is added to c_t through the combination of two gates, i_t, which decides the information to updated and the candidate values, c'_t. Finally, c_t is updated, and the output is a filtered version of the cell gate modulated.

AE

An AE is a type of ANN that replicates input data $\mathbf{x}$ to the output of the network $\mathbf{x}'$ with a minimum error. This mechanism allows extracting the most representative relationships from the data in its latent spaces, the so-called codes. AEs have an encoding function, $f_\theta(\cdot)$, which is the portion of the ANN that extracts knowledge in its codes $\mathbf{h} = f_\theta(\mathbf{x})$, and the decoding function, $g_{\theta'}(\cdot)$, in charge of reconstructing the input, $\mathbf{x}' = g_{\theta'}(\mathbf{h})$. The components of an AE can be appreciated in Fig. 5.2.

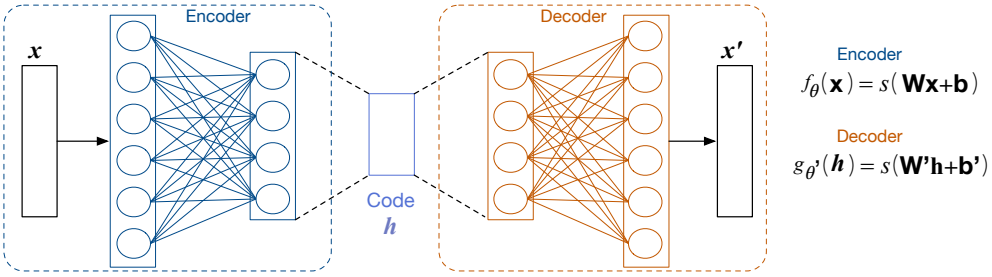


Fig. 2.19: Structure of an autoencoder with three hidden layers.

The learning mechanism of an AE follows the same as a standard ANN, except that the loss function varies. In this case, the RMSE is computed as

$$\mathcal{L} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} \|\mathbf{x}_i - \mathbf{x}'_i\|^2}, \tag{2.27}$$

where $\mathbf{x}_i$ represents a sample i and N is the total samples in a dataset.

2.4 Chapter summary

This chapter reviewed the components related to clinical data analysis. The mechanisms of clinical data manipulation in a KDD environment were addressed. Within this framework, we delved into the challenges associated with clinical data. These challenges are related to the volume of information and missing records in the data. Later on, the most widely used ML-based knowledge extraction mechanisms in the clinical environment were presented and those specialized in DL were discussed in more detail. In the

following chapters, we will show how these ML approaches can be used to solve the challenges posed by focusing on chronic kidney disease to get closer to data-driven medicine. Specifically, we will discuss the improvement of mortality prediction models in ESRD patients, present a DL-based imputation mechanism, and finally, a framework to perform TL and support mortality prediction in ESRD.

3

Enhancing Mortality Predictive in End-Stage Renal Disease by means of Deep Leaning

In this chapter, we begin to study the predictive models of mortality in ESRD and the influence of the application of DL on them. Thus, we started with collecting data from a nephrology unit and integrated various sources of information to generate a dataset. These include historical diagnoses, laboratory tests, and data from hemodialysis (HD) sessions. Thus, by combining these data with ML and DL techniques, we aim to improve the predictors of mortality for this pathology. We had a cohort of 261 patients and their temporal evolution throughout CKD. In this way, we exploit the predictive capacity of temporal dependencies through LSTM networks. Their performance is compared with learning models fed with different groups of attributes, with the twofold purpose of comparing the performance of groups of attributes chosen by the expert staff and those found by an ML approach such as recursive feature elimination (RFE). With this study, we have identified subsets of variables that perform very similarly to the use of all of them and allow us to identify which factors may have the most significant influence on mortality. Mortality predictors based on DL provide evidence that predictors based on EBM can be improved by combining different data sources and taking advantage of temporal relationships through LSTM networks. The main contributions of this chapter are as follows:

- ML approaches can reveal causal relationships in variables not explored before by the expert staff.
- The massive use of attributes together with DL approaches improves predictive models of mortality in ESRD.

- The integration of a ML pipeline may lead to a paradigm shift in analyzing predictive factors for mortality in ESRD.

3.1 Introduction

CKD represents an epidemiological problem, USA 11% and Spain 9.2% in the adult population [Mar+14]. According to the World Health Organization (WHO), it has an indirect impact on the morbidity and mortality of the global population, increasing the mortality risk of the deadliest diseases [LTS18; RK15]. CKD is closely related to cardiovascular (CV) risk, which is responsible for the highest mortality, especially on the ESRD, where death from CV is one of the leading causes [Mar+14].

The most widely used way to detect the risk of suffering these kinds of pathologies is based on EBM, which is translated into best practice guidelines, such as the American Heart Association/American College of Cardiology (ACC/AHA)[Gof+14], QRISK2 [Hip+08], Framingham [DAg+08a], or Reynolds [Rid+07]. They are based on assuming linear relationships between risk factors and events. Nevertheless, applying more sophisticated algorithms that use non-linear relationships and offer better performance in predictive models is still an open issue.

Thus, in the era of ML and DL, it is possible to generate complex models supported by large amounts of data [DO02; AAQ17]. Moreover, large-scale studies have begun to be described with ML to establish a prognosis of mortality in the general population using routine clinical data [Mil+19; Ros+16; Çel+14; Wan+15]. However, those that exist in ESRD use approaches based on classical statistics [Mau+08; Bed+00; Liu+10; Cou+09] and some of them present a doubtful benefit [Ote+12].

There are few studies where ML techniques are applied to CKD. Salekin [SS16], and Abdullah [Alm+19] detect CKD using different classifiers (SVM, k-Nearest Neighbors, RF and ANN), Doi [Doi+15] trains logistic regression to predict mortality in patients starting with hemodialysis, and Titapiccolo [Ion+13] stratifies cardiovascular risk with RF. Predictive models of mortality using ML are even scarcer in the ESRD population. Akbilgic [Akb+19] used

RF to predict mortality from one month to one year with an AUROC of 0.736. Thus, it is observed that there is a considerable margin for applying techniques that can benefit from both data complexity and the evolution of the patient's disease in ESRD for the improvement of mortality models and thus be able to support medical decisions that are reflected in the trajectory of the disease.

This chapter aims to present an exploratory analysis of the potential of DL by exploring heterogeneous variables and exploiting the follow-up of patients in ESRD through LSTM-ANN to improve predictive models of mortality. The predictive capacity of several collections of variables is explored. Such groups are chosen by the expert staff and automatically by ML techniques to evaluate the incidence of such features for learning models. This study also takes advantage of the number of samples generated by the continuous monitoring of patients to propose predictive models of short-term mortality, which have not yet achieved an AUROC higher than 0.736. This study points to the potential benefits of ML approaches to assessing the medical staff with ESRD patients. It encourages the development of more robust models using specialized ANNs as a predictive mechanism.

3.2 Materials and methods

This retrospective study was carried out on a homogenous cohort of 1178 HD patients from a single center with a reference population of almost half a million inhabitants. Of 1178, it was possible to extract information from 537 deceased patients, and of these, 261 provided the necessary data. These data were taken from the Information System of the Parc Tauli University Hospital, from the HD Unit at the Nephrology Department from 2007 to 2018. This project passed through the ethics committee (Code 2018/508) and was subsequently anonymized, following the usual protocol. Inclusion criteria was being of legal age ($>$ 18years). The available data include diagnoses, laboratory tests, and variables from HD sessions. The exposure period was from the moment the patient's information was registered in ESRD in the hospital's information system until the patient's death.

The predictive capacity of the data was exploited using the temporal dependencies that the follow-up of patients may have. Thus, a KDD process was followed, selecting the data by the expert staff, then the attributes were preprocessed, and finally, the predictive models were generated in two stages. Due to its easy-to-tune and computational cost, the first one uses RF to find the most important variables and set a baseline performance for more sophisticated algorithms. The second stage has the twofold purpose of exploiting temporal dependencies through LSTMs and analyzing the impact of sets of variables, including the ones found in the previous stage, groups of variables chosen by the expert staff, and using all the available ones. All the necessary steps to carry out the prediction of mortality for patients in ESRD can be seen in Fig. 3.1 and are described below.

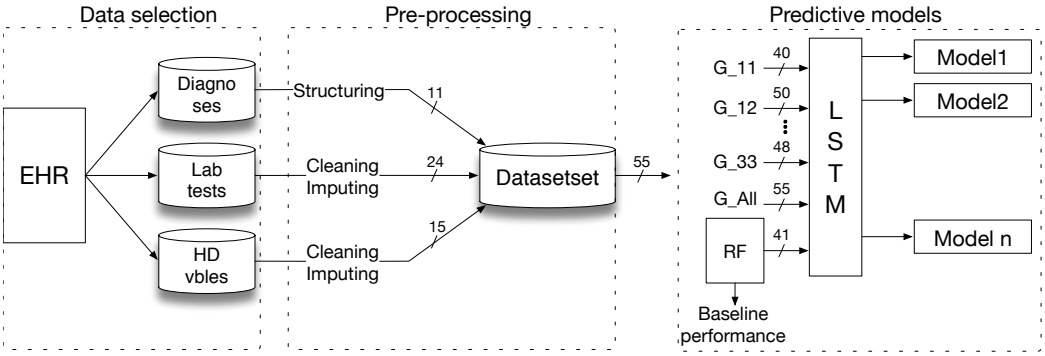


Fig. 3.1: Framework for developing predictive models in ESRD, G_{11} to G_{33} refer to set of variables ranked by their importance based on the expert staff experience. In pre-processing stage some features were generated based on 1-hot encoded for categorical features.

3.2.1 Data selection

Variables from the history of diagnoses, laboratory tests, HD sessions, and demographics are used to develop the predictive models. The outcome to predict is the mortality of patients. The variables are filtered based on their percentage of MV. Variables with more than 43.2% of MVs are discarded. Thus, the selected features can be appreciated in Table 3.1. Next are described the most relevant sources of information for this study.

Tab. 3.1: Selected variables from the data sources. The outcome is coded according to the death date of the patient. SBP and DBP refer to systolic and diastolic blood pressure, HR to heart rate and Temp to temperature.

Laboratory tests	**hemodialysis**	**Diagnoses**
Calcium	Acc weight	Arteriopathy
Creatinine	Average flow	Cardiopathy
Ferritin	Blood vol dia	Diabetes
Glucose	DBP post HD	Enteropathy
Haemoglobin	DBP pre HD	Fracture
Haemoglobin	Dry weight	Haemorrhage
HDL cholesterol	HD time	Hepatopathy
Hematocrit	HR post HD	Hypertension
Iron	HR pre HD	Infection
KTV	Hypotension	Neoplasia
LDL cholesterol	SBP post HD	Pneumopathy
leucocytes	SBP pre HD	
Lymphocytes	Temp post HD	
Monocyts	Temp pre HD	**Demographics**
Neutrophil	Vascular access	Age
Phosphorus		Sex
Platelets		
Potassium		
PTH		**Outcome**
Reticulocytes		Months to decease
Sodium		
Total cholesterol		
Total proteins		
Triglycerides		
Urea		

Diagnoses

Diagnoses refers to the historical hospital admissions that a patient has had. Each entry is associated with some particular diagnoses determined by examinations and evaluations of medical staff, which is encoded using ICD-9.

Laboratory

Laboratory tests are all the associated attributes collected from hematology, biochemistry, or some ESRD-related hormones. Such samples are stored as laboratory events. Some of them are taken with more or less periodicity. For instance, the most regular is the hemoglobin, measured every month, while proteins and PTH are measured every four months. Other measurements like immunology or tumor markers are taken more exceptionally.

HD variables

HD variables are commonly taken during the HD sessions, 3-4 per week. Some variables are recorded at the beginning and the end of the HD session. Registered information includes the type of vascular access, duration of the session, episodes of hypotension, and other variables taken from the hemodialysis machine, such as dry weight, temperature, systolic and diastolic blood pressure, heart rate, and average flow, among others.

3.2.2 Data pre-processing

In general, as mentioned in the previous chapter, EHRs rarely have an appropriate format for feeding learning models. Thus, it is necessary to carry out exploration and preprocessing of such records to optimize the knowledge extracted from them. The aforementioned is carried out through the cleaning and imputation of MVs. Below are the problems found in the samples.

- Data structure;
- Incorrect values in variables;
- MVs.

Initially, the information has to be structured. Thus, based on expert knowledge, the diagnoses were grouped into 11 general ones. Then, they were categorically encoded so that they became new variables for the final data set. In Fig. 3.2, this transformation can be appreciated. Then, the three

information sources are combined based on the measured data. Finally, the follow-up of patients was summarized into one-month records, i.e., using the mean of variables in case of having more than one sample per month.

Patient_id	D_ad	C_Diag_1	C_Diag_2	C_Diag_3	C_Diag_4	...
14785899	13/12/11	I351	250.40	585.5	V45.1	...
14785899	02/06/08	I6529	428.0	I4891	250.40	...
14785899	07/01/14	707.19	858	V45.1	Z8611	...
14785899	17/10/14	I2510	I472	I120		...

Patient_id	D_ad	ARTERIOPATHY	CARDIOPATHY	DIABETES	...	INFECTION
14785899	13/12/11	0	1	0	...	0
14785899	02/06/08	0	0	1	...	0
14785899	07/01/14	1	0	1	...	1
14785899	17/10/14	0	1	1	...	0

Fig. 3.2: Initially, each entry is associated with a series of diagnoses. In the new scheme, the most important diagnoses are selected and coded using one-hot encoding.

To correct the outliers and impute MVs, the acceptable ranges of the variables in laboratory tests and HD sessions were decided by the expert staff. Outliers of variables are identified and replaced with MVs to avoid losing the rest of the information of encounters with the MVs. Then, they are processed in two stages. The first one is based on the individual imputation of the variables of each patient using second-order interpolations to preserve trends in the evolution of the patient. MVs are imputed for patients without samples in some variables in the second stage based on average values of the respective attribute from patients in training set without MVs.

3.2.3 Learning models

Models in the literature dedicated to the study of CKDs have been shown to have performance that can be improved. RF is the best performing algorithm in this type of cohort. However, those learning algorithms do not consider the temporal component that exists in the evolution of the pathology. Thus, RF is used to establish baseline performance and also quantifies the features'

importance for the final predictor. Then, LSTM-ANN is applied to benefit from the temporal component mentioned above.

Feature selection-RF

RF combines predictions based on decision trees [BL01]. They are trained with random subsets of data D_n. Branches of the decision trees are generated based on the calculation of the impurity of their features through the *Gini* index,

$$G(D_n) = 1 - \sum_{i=1}^{m} {p_i}^2 \tag{3.1}$$

where *m* is the number of classes (2 in our case, dead or alive), and p_i is the relative frequency of class *i* in a given branch of the tree. Initially, $G(D_n)$ is calculated for all the possible combinations of features and for finding the value used to split the tree's branches. Then, the combination that achieves the lowest value of $G(D_n)$ is chosen as it represents the best possible value in D_n at the classification nodes in the tree. The same procedure is repeated in subsequent branches up to a specified depth. In an RF approach, several trees are computed and fed with subsets of the data. Finally, the outcome produced by most of the trees is taken as the final decision (see Fig. 3.3).

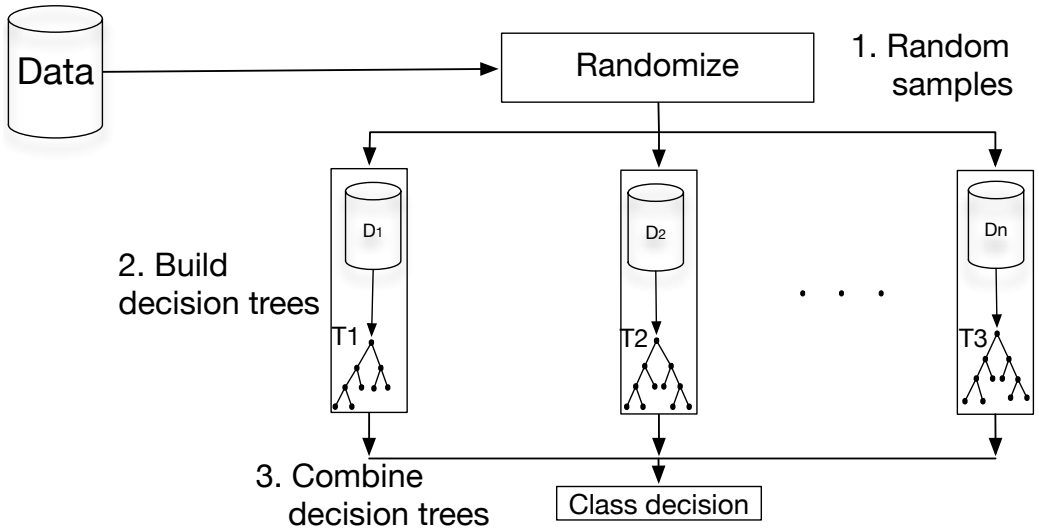


Fig. 3.3: Random forest data flow, at the end the class decision is made by voting of each tree.

On the other hand, as was explained for the entropy in decision trees, the *Gini* index allows quantifying the importance of the features. This characteristic is used in this work to find the most relevant variables, more robustly, for predictors by combining a recursive feature elimination (RFE) [DME18] approach with RF. The traditional way to find the importance of features is to relate them individually with the outcome without considering the interactions between variables. RFE solves this issue by generating several predictors iteratively. Thus, in each iteration, a predictor offers a performance measurement and the ranking of features. In the next iteration, the less important feature is eliminated and the new predictor will yield another performance and a new ranking, and so on.

Predictive model-LSTM

In the case of the prediction of mortality of patients in ESRD, the objective is to classify data collected during a period of n_{data} months to be able to determine if the patient will be alive or not after n_{pred} months. Thus, driven by a considerable number of training samples, an ANN can learn an optimized non-linear function in an iterative process to minimize the input and output error. Thus, to carry out mortality prediction in ESRD, LSTMs are used to exploit temporal dependencies in the follow-up of the patients. LSTMs are fed with concatenated vectors that contain the evolution of n months, and the prediction is carried out to p months. Fig. 3.4 illustrates the follow-up of a patient during *m* months, from the first encounter with the hospital's system to the deceased event. The follow-up is structured into samples, taking information of *n* months of evolution. Then, using the timestamp of the samples and date of deceased, *d* in the figure, the moths to the death event of the structured samples are computed. Thus, the binary target of the generated data depends on the prediction range using the rule,

$$f(t_d) = \begin{cases} 0, & \text{if } t_d > p \\ 1, & \text{otherwise} \end{cases} \tag{3.2}$$

where p is the prediction range, t_d is the time to the death event. '0' and '1' indicate the class sample, alive and deceased, respectively.

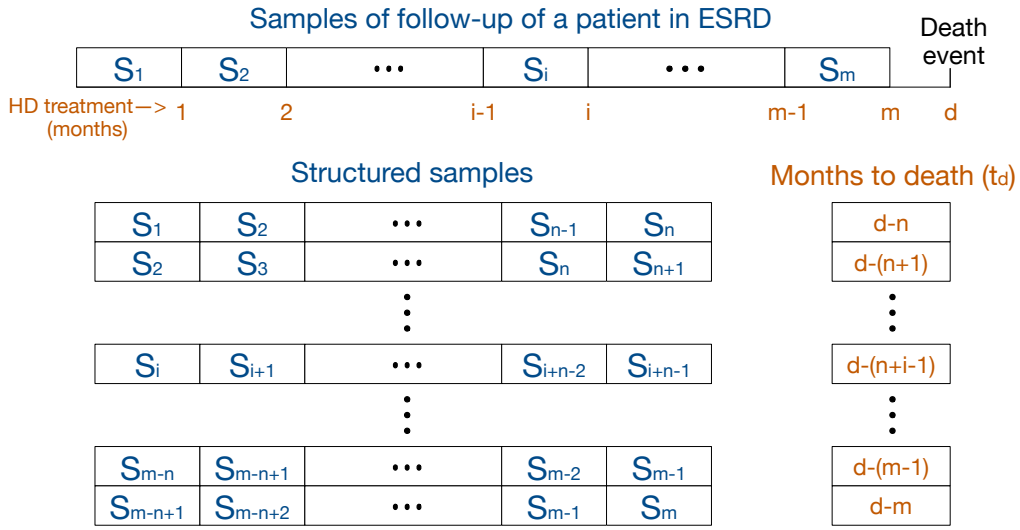


Fig. 3.4: Sample structuring from follow-up of a patient with m months in HD treatment to the death event, d.

3.3 Results

The samples for this analysis were extracted from 261 patients. Table 3.2 shows the description of the population. Because the duration of HD treatment varies across the cohort, each patient generates a different number of monthly samples. In total, 8394 samples were extracted. Thus, mortality is predicted to 1, 2, 3, 6 and 12 months in this work. Then, five datasets with the same data but different targets, after applying the transformation in Eq. 4.1, are generated. Fig. 3.5 shows the mortality trajectories for patients in the training and test sets.

For models development, patients were split into training and test sets (80-20%). The training set was divided into 5-folds for cross-validation, see Fig. 3.6. This approach makes it possible to find the hyperparameters for RFE-RF and LSTMs. Such parameters are the ones that can be calibrated manually. For RFE-RF, the number of trees, depth of the decision trees and splitting criteria. For LSTMs, the number of cells, neurons per cell, LR, among others. Then, with the hyperparameters fixed, the parameters of the network (weights of the LSTMs) are computed and five different models, $M1, M2, \ldots, M5$ in Fig. 3.6, from the 5-folds, are obtained as a result. The evaluation is done in the initial test set.

Tab. 3.2: Cohort description. Variable Samples/Patient contains the information about the number of samples that the patients generate. For diagnoses, Number of patients column represents the total of patients with a specific diagnose. VA refers to vascular access, SBP and DBP to systolic and diastolic blood pressure.

Feature	Units	Patients	MV (%)	Mean	Std	Min	Max
Age	Years	-	0.0	71.41	10.69	24.00	91.00
Sex (Women)	-	104	0.0	-	-	-	-
Sex (Men)	-	157	0.0	-	-	-	-
Samples/Patient	-	-	-	26	22	1	116
Calcium	mg/dL	261	10.8	9.10	0.69	6.30	13.00
Creatinine	mg/dL	261	25.0	6.80	2.30	0.30	15.50
Ferritin	ng/mL	261	28.1	472.1	368.32	8.10	6590.00
Glucose	mg/dL	261	25.7	123.30	67.85	13.00	1370.00
Haemoglobin	g/L	261	29.2	6.21	1.26	4.10	13.60
HDL cholesterol	mg/dL	261	18.3	43.73	14.60	4.40	115.60
Hematocrit	L/L	261	1.0	349.990	0.04	0.17	0.49
Hemoglobin	g/L	261	1.1	111.69	14.21	46.00	161.00
Iron	μg/dL	261	38.1	59.44	26.70	10.00	340.00
KTV	mL/min	261	17.3	1.43	0.28	0.42	02.09
LDL cholesterol	mg/dL	261	18.9	83.40	33.20	8.00	240.00
Leucocytes	x$10^9/L$	261	1.0	7.63	4.97	1.25	11.3
Lymphocytes	x$10^9/L$	261	5.8	1.50	0.76	0.22	12.74
Monocyts	x$10^9/L$	261	5.8	0.56	0.22	0.03	2.69
Neutrophil	x$10^9/L$	261	5.8	5.24	2.29	0.22	7.25
Phosphorus	mg/dL	261	26.1	4.33	1.39	0.20	11.80
Platelets	x$10^9/L$	261	1.1	223.37	83.17	14.40	1067.00
Potassium	mEq/L	261	35.0	4.95	0.80	0.30	8.90
PTH	pg/mL	261	28.3	228.05	189.17	6.00	3264.00
Reticulocytes	x$10^9/L$	261	28.4	5.37	2.69	0.23	35.23
Sodium	mEq/L	261	31.5	138.66	3.59	121.00	198.00
Total cholesterol	mg/dL	261	38.1	149.98	39.41	45.00	432.00
Total proteins	g/L	261	27.6	66.02	6.84	28.5	96.00
Triglycerides	mg/dL	261	18.1	140.49	107.92	20.00	2673.00
Urea	mg/dL	261	43.2	102.40	51.12	20.20	317.20
Accumulative weight	Kg	261	21.7	1.95	0.77	-3.05	4.44
Average flow	mL/min	261	16.2	290.28	34.48	200.00	414.55
Blood vol dia	mL/min	261	12.0	65.08	10.52	40.00	98.43
DBP post HD	mmHg	261	10.4	65.34	10.22	40.00	105.61
DBP pre HD	mmHg	261	10.5	64.44	10.62	40.00	106.08
Dry weight	Kg	261	0.9	66.78	15.24	31.29	149.63
HD session time	Hours	261	0.0	3.73	0.35	3.50	7.30
HR post HD	BPM	261	10.6	75.59	11.49	41.00	122.00
HR pre HD	BPM	261	6.6	73.19	10.57	42.00	121.17
Hypotension	Cases/month	261	0.0	2	4	0	24
SBP post HD	mmHg	261	13.3	138.11	22.7	57.00	205.00
SBP pre HD	mmHg	261	6.3	137.31	22.19	56.07	218.60
Temp post HD	$^{\circ}C$	261	16.9	35.58	0.33	33.00	38.20
Temp pre HD	$^{\circ}C$	261	11.6	35.52	0.34	33.85	38.00
Arteriopathy	-	177	0.0	-	-	-	-
Cardiopathy	-	241	0.0	-	-	-	-
Diabetes	-	204	0.0	-	-	-	-
Enteropathy	-	94	0.0	-	-	-	-
Fracture	-	9	0.0	-	-	-	-
Hemorrhague	-	6	0.0	-	-	-	-
Hepatopathy	-	18	0.0	-	-	-	-
Hypertension	-	223	0.0	-	-	-	-
Infection	-	102	0.0	-	-	-	-
Neoplasia	-	79	0.0	-	-	-	-
Pneumopathy	-	115	0.0	-	-	-	-
VA (AVF)	-	168	0.0	-	-	-	
VA (Catheter)	-	164	0.0	-	-	-	-
VA (Graft)	-	6	0.0	-	-	-	-
Mortality	Months	-	0.0	25.52	21.89	1.00	116.00

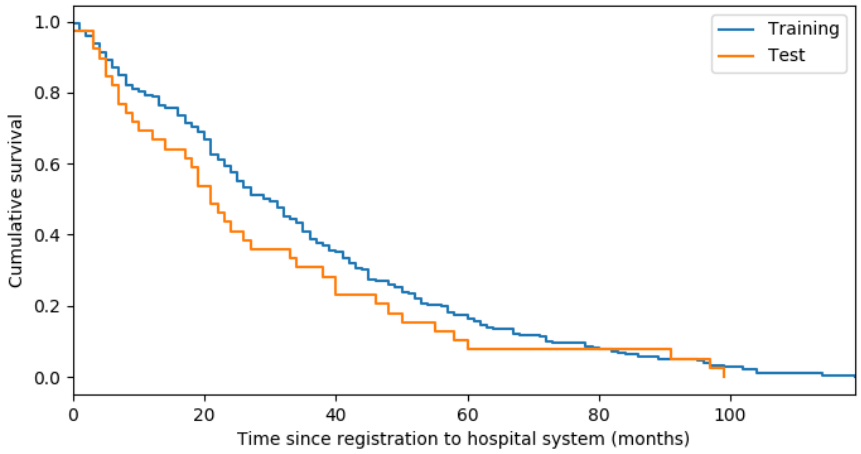


Fig. 3.5: Kaplan Meier mortality model for training and test set.

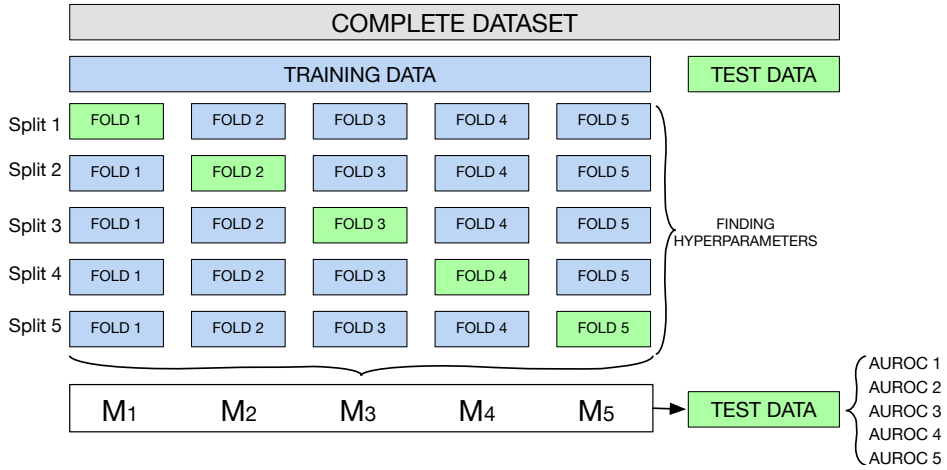


Fig. 3.6: Cross validation with 5-folds. Test Data is only used when the hyperparameters are found.

To estimate the performance of the classifiers in the test set, AUROC was used. It measures the area under the graphic representation of the general accuracy, showing the variation of the sensitivity and specificity of a binary classifier when the decision threshold varies. The metric takes values between 0 and 1, with 1 corresponding to the perfect classifier.

3.3.1 Feature selection-RF

Our first experiment studies the importance of groups of features. One of them is the group found by combining RF with the RFE approach. The

optimal hyperparameters for RF were 103 trees, maximum depth of 3, using the *Gini* index for splitting the nodes and calculating the importance of features. For the RFE approach, the 5-folds were used to find the best features more robustly. With the approach, it was found that 42 features offered the best performance for all the predictors. AUROCs of 0.737, 0.714, 0.712, 0.668 and 0.615 were the baseline performance predicting mortality to 1, 2, 3, 6 and 12 months, respectively. Fig. 3.7 illustrate the AUROC as a function of the number of considered features for the prediction of mortality to one month.

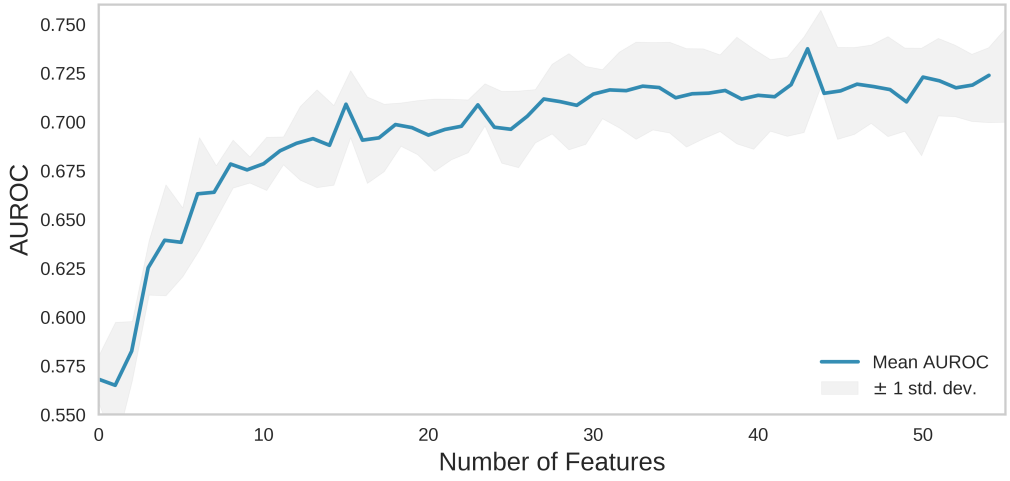


Fig. 3.7: Recursive feature selection, with 5-folds cross-validation, using RF as learning model.

The features not considered by RF-RFE were: cardiopathy, enteropathy, haemorrhage, hepatopathy, hypertension, neoplasia, pneumopathy, fracture, infection, and the type of vascular access.

3.3.2 Predictive model-LSTM

In the second experiment, we consider a more powerful model based on LSTMs. After parameter optimization, we found that the best configuration was using an LSTM with two cells and with 750 and 500 units, respectively. We used ADAM optimizer with LR = 0.001 and L2 regularization with $\lambda = 0.001$. Then, the LSTM approach is evaluated in several groups of

variables chosen by the experience of the expert staff, the group of variables found by RFE-RF, and all the available ones. Table 3.3 shows the importance level of both laboratory and HD variables as determined by the experience of the hospital expert staff.

Tab. 3.3: Ranking of features chosen by the experience of the expert. Their importance are marked from 1 to 3, being 3 the less important features. VA refers to vascular access.

Laboratory	Importance	HD variables	Importance
Calcium	1	HD time	3
Creatinine	3	HR post HD	1
Ferritin	2	HR pre HD	1
Glucose	3	Hypotension	1
Haemoglobin	1	SBP post HD	1
HDL cholesterol	2	SBP pre HD	1
Iron	3	Temp post HD	3
KTV	1	Temp pre HD	3
LDL cholesterol	2	VA (AVF)	1
Leucocytes	2	VA (Catheter)	1
Lymphocytes	2	VA (Graft)	1
Monocytes	2		
Neutrophil	2		
Phosphorus	1		
Platelets	3		
Potassium	2		
PTH	1		
Reticulocytes	2		
Sodium	1		
Total cholesterol	2		
Total proteins	3		
Triglycerides	2		
Urea	1		

Fig. 3.8 shows ROC curves comparing the groups of variables using four months to feed the LSTM and predicting mortality to 1 month. As an illustration, Group_12 considers laboratory variables with an importance label of 1 and HD session variables with an importance level of 2, and Group_RFE refers to the ones found by RFE-RF. Note that diagnosis variables (11 in total) are included in all cases. In this figure, we can see how the performance offered by the models fed with the variables found by the RFE-RF approach is enhanced once the LSTMS-ANN is applied.

Finally, in Fig. 3.9 we test the performance of our algorithm by considering: i) all variables; ii) HD data only and iii) diagnosis and laboratory data only.

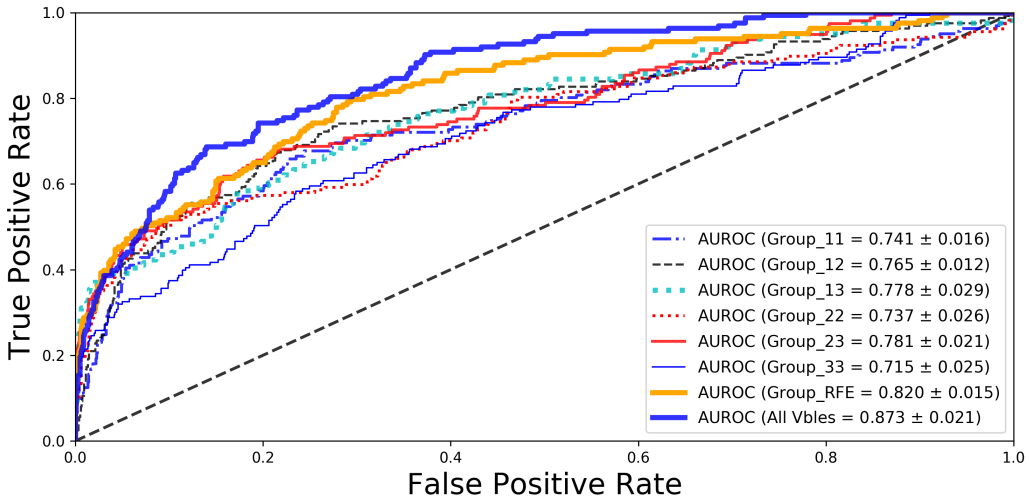


Fig. 3.8: Performance comparison between best features found by RFE-RF, the combinations of features chosen by expert staff and using all the available information. Group_11 is inferred from the combination of the most important analytics with the most important HD variables to Group_33, the least significant ones.

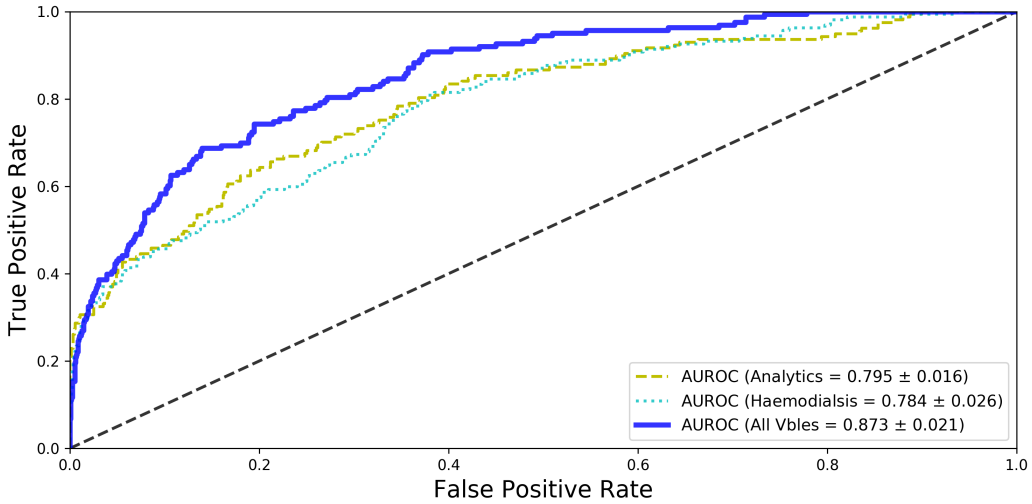


Fig. 3.9: Comparison of performance using laboratory tests and diagnoses, all the available variables and just variables taken during hemodialysis sessions.

3.4 Discussion

This chapter explored how DL can help in the study of ESRD. After the experiments conducted, in this case, focused on the evaluation of mortality, the lessons learned are: i) we can improve model accuracy w.r.t. the other works in the literature; ii) including knowledge expert not always leads to better models and iii) solutions can guide the research in a specific field by revealing possible causal relations not explored before, possibly far from human intuition. Table 3.4, includes a performance comparison in terms of AUROC with the existing solutions in the literature. Although one-year mortality does not exceed that stated in the literature, the improvement in short-term mortality grows to 4% if we reduce the prediction time to 3 months. When we compared our approach to these other works, we realized that we combined three sources of data, i.e., diagnosis, laboratory and HD data, being not the case in the available works. Most of them use either laboratory and diagnosis data or HD session data. Fig. 3.9 shows that the inclusion of all variables improves the AUROC at least by 11% in AUROC.

Tab. 3.4: Comparison methods in literature with proposed one.

Reference	Mortality	Population	Prediction algorithm	AUROC (CI 95%)
[Mau+08]	1 year	5738	Logistic regression	0.670 (0.668-0.675)
[Akb+19]	1 month	27615	Random forest	0.736 (0.715-0.757)
	3 months			0.764 (0.754-0.774)
	6 months			0.760 (0.747-0.775)
	1 year			0.757 (0.746-0.769)
[Wag+11]	3 years	5447	Cox	0.730 (0.700-0.760)
	7-12 months		proportional	0.698 (0.679-0.717)
	13-18 months		hazards	0.717 (0.696-0.737)
	19-24 months			0.670 (0.646-0.694)
[Iss+14]	3-36 months	62	Hazard ratio	0.696
Proposed	1 month	261	LSTM	0.873 (0.871-0.876)
approach	2 months			0.813 (0.811-0.815)
	3 months			0.798 (0.796-0.800)
	6 months			0.752 (0.751-0.753)
	1 year			0.720 (0.703-0.737)

In order to study how considering knowledge expert influences the performance of algorithms, expert staff labeled HD and laboratory data according to their importance level, being 1 the highest level and 3 the lowest (see Table 3.3). Accordingly, in Fig. 3.8 we tested our model with several combinations of the subsets of variables. We could expect to achieve the best

possible performance by using level 1 laboratory data together with level 1 HD variables, i.e., Group_11 (recall that diagnostic data is included in all cases). However, the performance achieved is similar to Group_33, and the inclusion of all variables boosts the AUROC value in 9%. In other words, expert knowledge is undoubtedly relevant, but it is also important to explore beyond it.

Finally, ML approaches can also help the research of physicians by revealing causal relations possibly not explored before. In Fig. 3.7, we tested how an automatic feature selection tool such as RFE may help. In this case, 42 features gave us the best classification performance using an RF approach, where the majority of the diagnoses were excluded. When we consider this selection as input to our LSTM solution, the performance is close to the best one, obtained with all the features. Therefore, physicians can explore the subset of variables selected, reduce or increase it as far as performance is sustained (see Fig. 3.7) and investigate the importance and effects of the chosen features. However, it should be noted that RF-RFE did not considered most of the diagnoses or the type of vascular access. This evidence could suggest that those not considered important variables could lead to new medical research.

3.5 Conclusions

The work presented in this chapter demonstrates the potential of the massive use of variables and ML techniques to improve predictive mortality models in ESRD. We designed a baseline predictor and feature selector using an RFE-RF approach. Then we improve it using LSTM strategy that exploits temporal dependencies in the data. We conclude that thanks to considering diagnostic variables and laboratory and HD session data, we could improve performance in predicting mortality in the ESRD patient by at least 4% w.r.t. existing works for short-term mortality. Furthermore, results show that expert knowledge contributes to the analysis, but we shall not limit our algorithms to it. In our experiment, the best performance achieved by the groups chosen does not exceed the RF-RFE. Therefore, ML methods like the ones explored here can provide feedback to the experts, improve our knowledge, and lead to a

change in the paradigm in the analysis of predictive factors in mortality in ESRD.

4

Multiple Imputation Using the Average Code from Autoencoders

In this chapter, we cover the second challenge of this dissertation, which is related to the imputation of MVs in clinical data. For this purpose, we group 4 clinical datasets with different pathologies widely used in the literature and two others that are frequently used as benchmarks in an imputation environment. The new imputation alternative proposed in this chapter integrates a classical MI-based approach within a DL environment. This concept integrates MI within the latent spaces of an EA, where what we call the average code is computed. This code is used to reconstruct non-existing information better. This mechanism makes it possible to decrease the bias generated by the imputation of non-existing information. With this proposal, a better reconstruction of the different data sets has been demonstrated.

The main contributions of this chapter are as follows:

- Provide a novel alternative to compute a more robust representation of latent spaces by computing an average latent representation.
- Integrate the MI paradigm into latent spaces of deep ANN.
- Improve the imputation of missing values by computing the average code for datasets independently from the volume of data.

4.1 Introduction

Missing information is inherent to data used in pattern recognition, data mining, and machine learning applications. The presence of MV affects the quality of such applications and may add biases to experiments [Gia+18]. In the clinical domain, this is a recurrent problem. Electronic health records such as laboratory tests, clinical observations and bedside monitoring usually present MVs in their registers. The integration of such data, the omission of information by patients, data acquisition equipment errors, and measurements with different sampling periods are the most representative sources of MVs in the medical domain [Cis+13; Mir+16].

The alternatives to handle this issue include removing records with MVs and applying methods to estimate them. The first option is known as complete case analysis and is commonly used in clinical studies [Cis+13]. This approach has the drawback of considerably reducing the amount of data, adding bias to the experiments since it analyses only the complete examples in the dataset [Ste+09]. In contrast, imputation attempts to replace missing information with the twofold purpose of extracting knowledge from incomplete examples and reducing bias in clinical studies [Ste+09].

Imputation methods replace the MVs considering either one value or multiple estimates for an MV. Relying on imputing by one value underestimates the variance and does not consider data uncertainty [CR89; Ste+09]. Moreover, MI [CR89] was designed to address this concern by considering several estimates for a single MV. However, the challenge in MI lies in the choice of the estimative model [GW18]. Statistical models for MI are based on estimates that consider only linear relationships in the data. In several scenarios, they cannot handle large datasets and are limited when there are different types of data and MVs patterns. In contrast, such limitations may be exploited by methods based on DL.

DL techniques have shown an exceptional ability to exploit complex relationships in large datasets [Chi+18]. Such relationships are extracted in latent representations in the hidden layers of ANN. Promising methods in imputation in the literature are based on GANs [Goo+14; YJV18] and AEs [BM17; Mac+19; GW18]. However, since GANs are based on two competing

ANN, they are difficult to tune and often present convergence problems [AKK19]. In contrast, AEs learn unsupervised functions that encode the input into a latent representation of data and then use a decoder function that reconstructs such representation to match the input. This mechanism makes it possible to extract the most relevant information from the data, even from those examples that present MVs. Thus, it is necessary to carry out an initial imputation to include such examples. Most authors use a constant value to perform this initial imputation without considering that the learning models can memorize the data they were fitted with. Thus, choosing adequate values to perform this initial imputation is a challenge to minimize the bias they add.

On the other hand, the application of AEs, jointly with the MI approach, has shown promising imputation solutions [GW18]. In the method proposed in [GW18], several copies of the data are estimated by AEs. The MI mechanism is applied once the AE reconstructs the latent representations. Thus, motivated by the ability to extract knowledge that AEs have and the inclusion of uncertainty that MI provides, in this chapter, we present a novel alternative to impute MVs based on the adoption of the MI paradigm into latent representations of data through AEs. This integration allows the combination of several latent representations of the data into the so-called average code (AVG code). This combination generates a more robust representation of data. Once the AVG code is decoded and missing information is better imputed, the complete information may support learning tasks in specific domains.

4.2 Methods

This section presents all the necessary components to carry out the proposed method. Initially, the imputation problem is formally introduced, followed by the description of MI and the learning models on which the proposed method is based. Finally, the proposed method is presented.

4.2.1 Imputation problem

Let $\mathbf{X} \in \mathbb{R}^{s,f}$ be a dataset, with $s = 1, \ldots, S$ examples and $f = 1, \ldots, F$ features. The elements of $\mathbf{X}$ are denoted by $x_{s,f}$. Each sample is denoted by $\mathbf{x}_s = [x_{s,1}, x_{s,2}, \ldots, x_{s,f}, \ldots, x_{s,F}]$. Then, a MV indicator $\mathbf{m}_s = [m_{s,1}, m_{s,2}, \ldots, m_{s,f}, \ldots, m_{s,F}]$ is associated to the examples $\mathbf{x}_s$ and tracks the registers that are missing. Each element of matrix $\mathbf{M}$ is constructed as follows:

$$m_{s,f} = \begin{cases} 1, & \text{if } x_{s,f} \text{ is missing} \\ 0, & \text{if } x_{s,f} \text{ is observed.} \end{cases} \tag{4.1}$$

Thus, $\mathbf{X}$ can be divided into two components, observed and missing data, $\mathbf{X}_{obs}$ and $\mathbf{X}_{miss}$, respectively. $\mathbf{X}_{obs}$ contains examples without MVs in their features, while those examples with MVs are stored in $\mathbf{X}_{miss}$. Imputation aims to find a function, $f(\cdot)$, that best estimates MVs in $\mathbf{X}_{miss}$, which in turn minimizes bias added by the inclusion of information that did not exist before. This function can be generated based only on the $\mathbf{X}_{obs}$ or by including also $\mathbf{X}_{miss}$ in its estimation. The first case considers the distribution of the observed data, and the MVs of $\mathbf{X}_{miss}$ are replaced by values that best fit such distributions. When $\mathbf{X}_{miss}$ is included for the computation of $f(\cdot)$, it is necessary to perform an initial imputation. This imputation replaces MVs and works as seed values that change iteratively in the training process for the imputation model.

Including examples with MVs adds robustness to the models because it is possible to extract knowledge that is directly linked to the appearance and generation of MVs in the dataset. Fig. 4.1 illustrates the imputation problem in two stages. The first one is the generation of the function $f(\cdot)$. An initial imputation is performed to compute the function that best fits the data. In the second stage, $f(\cdot)$ is applied in new data $\mathbf{X}^*$ and imputes their MVs. In summary, a robust imputation should include information from the examples with MVs and an adequate imputation model that minimizes bias in the experiments.

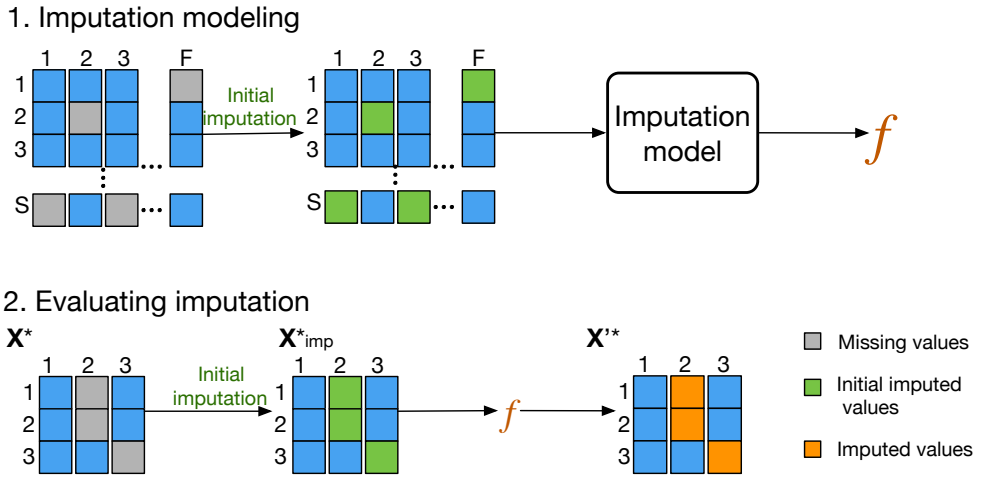


Fig. 4.1: Imputation scheme. The initial imputation is necessary to fit the model that reconstructs missing information.

4.2.2 Multiple imputation

MI creates several versions of a data set that are used to improve a single imputation. Each version may contain different estimates for the MVs. This imputation paradigm addresses the problem of uncertainty that exists when imputing an MV with a single estimation. The mechanisms to find imputation functions follow the same process mentioned in the imputation problem. In this case, N versions of the data are generated, and the initial imputation replaces MVs with slightly different values. Then, N imputation models generate estimates that are finally grouped and estimate the MVs of the dataset.

4.3 Proposed method

Motivated by the ability to represent complex relationships that AEs have and the solution that MI presents to handle the uncertainty problem, we propose to compute the AVG code of an AE as a mechanism for enhancing imputation. In our approach, we reinterpret the solution proposed in the work entitled Multiple Imputation using Denoising Autoencoder (MIDA) [GW18], by integrating MI in the latent spaces of an AE instead of at the output layer of the AE, as in MIDA. The proposed method consists of two

stages, the training of an AE and the imputation mechanism based on the AVG code.

Learning model

For the knowledge extraction process, an AE is trained. The dimension of the latent representations is smaller than the dimension of the input data. This stage differs considerably compared to MIDA. N AEs are trained in MIDA, and in our approach, just one is used to perform the learning task. To carry out the initial imputation, MIDA uses the average value to impute MVs. Imputing with constant values may cause conflicts in generalization for the learning model. The model may memorize the values with which the MVs were initially imputed. To solve this issue, we carry out the initial imputation based on random values that follow the distribution of the attributes of the observed values in the training data.

Additionally, as a regularization mechanism, the imputing values are changed in every epoch of the training process. The most representative categories are used as imputers for categorical variables, and these change in every epoch. An epoch is when the entire training sample is passed forward and backward through the AE only once. This iterative variation is carried out with the twofold purpose of preventing the models from memorizing the imputing data and guaranteeing a more robust representation of the data in the codes. Finally, the encoder and decoder are extracted from the trained AE. Fig. 4.2 illustrates the initial stage of the proposed method.

Imputation

At this stage, the MI approach is integrated into the latent space of the trained AE. The proposed method differs from the usual MI because the combination of information is not performed at the end of the estimation of the MVs, but in the latent space of the AE, as illustrated in Fig. 4.3. To perform the imputation on new data, $\mathbf{X}_{test}$, it is necessary to generate different copies of randomly imputed data. This imputation follows the same mechanism as the initial imputation described for the previous learning process. Then,

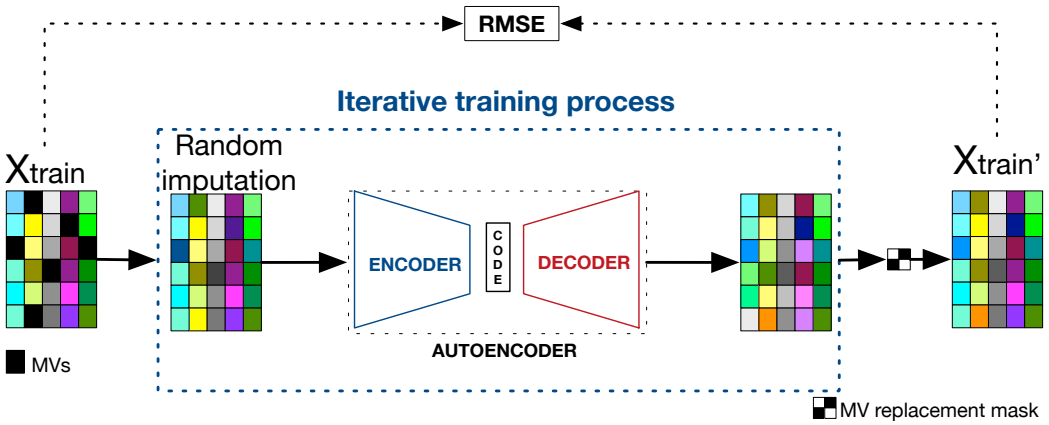


Fig. 4.2: Learning process of the proposed method. Missing values (MV) replacement mask contains the location of MVs in $\mathbf{X}_{train}$ and it is used to preserve the observed values of $\mathbf{X}_{train}$ and replace the MVs with values estimated by the AE.

the encoder function of the trained AE generates the codes. These codes are combined into an AVG code as follows,

$$\bar{c} = \frac{1}{N}\sum_{i=1}^{N} c_i. \tag{4.2}$$

The decoding function is applied to the AVG code. The final reconstruction, $\mathbf{X}'_{test}$, is the mixture of the observed data and the data that are tracked by the MV indicator.

4.4 Results

Six datasets have been used to test the imputation capacity of the AVG code. Diabetes, breast cancer, and liver datasets have been extracted from the UCI repository [DG17]. Spam and letter datasets have been included as they are widely used as benchmark datasets to evaluate imputation methods. Finally, in the datasets, we have included data extracted from the MIMIC-III database [Joh+16a] and that is related to CKD. From this massive database,

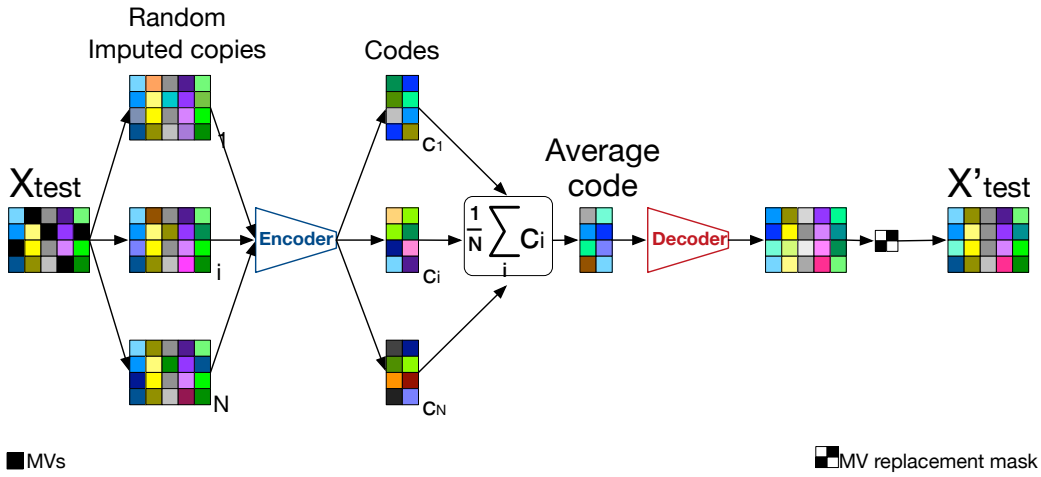


Fig. 4.3: Second stage of the proposed method. Encoder and decoder functions are extracted from a trained autoencoder. $\mathbf{X}_{test}$ are different data from training data and $\mathbf{X'}_{test}$ is the reconstruction considering a missing values (MV) replacement mask that contains the location of MVs in $\mathbf{X}_{test}$ and it is used to preserve the observed values of $\mathbf{X}_{test}$ and replace the MVs with values estimated by the decoder function.

those patients with acute kidney injury (AKI) were filtered based on the kidney disease improving global outcomes (KDIGO) clinical practice guideline [Ink+14]. All mentioned datasets have a mixture of categorical and continuous attributes. Table. 4.1 shows the dimension and amount of attributes of the datasets.

Tab. 4.1: Dataset used to compare the performance of the imputation mechanisms.

Dataset	Examples	Attributes
Diabetes	442	10
Breast cancer	569	30
Liver	579	9
Spam	4601	57
Letter	20000	16
AKI	56274	8

4.4.1 Experiments

The performance of the AVG code was compared with two imputation methods from the state-of-the-art, imputing MVs with MIDA and using an imputation alternative based on GANs [YJV18]. The imputation capacity of each

model is measured by computing the RMSE in the imputation stage for each model. The comparison of the three methods is carried out using the same data with 5-fold cross-validation. The same data for training and testing for the three methods ensure that the results are fair and generalizable to different data subsets.

The architecture and the hyperparameters used to train the state-of-the-art imputation models are proposed in such works. AEs have been trained with two hidden layers of $F/2$ and $F/4$ units for the AVG code, where F is the number of attributes of a specific data set. The inputs are standardized between 0 and 1 to facilitate the convergence of the models. To speed up the training, ADAM [KB15] optimizer was used with an $LR = 0.001$. A dropout [Sri+14] of 0.1 was applied as a regularizer to the hidden layers in the AEs. In addition to using conventional regularizers, early stopping [Pre12] was used to prevent the learning model from being overtrained and stop the training process when a model stopped learning.

The imputation capacity of the AVG code is evaluated in scenarios where the patterns of MVs vary. MVs are synthetically generated since the datasets do not contain MVs. To provide a wide range of comparisons, part of the information is eliminated in the experiments to generate MVs with ratios ranging from 10-60%. Ten copies of the data have been used and imputed with random values to generate the AVG codes of the experiments. Next, the mechanisms used to generate the synthetic MVs are covered.

MCAR

The first experiment, it is evaluated how the random appearance of MVs affects imputation. For this case, MVs are synthetically generated, varying the percentage of MVs from 10-60%. Such a scenario is the most typical in real-life datasets. In Fig. 4.4 it can be appreciated that the AVG code has a lower reconstruction error than state-of-the-art solutions. Additionally, it can be seen that the models based on GANs are very volatile to the change in the MV rate with MCAR.

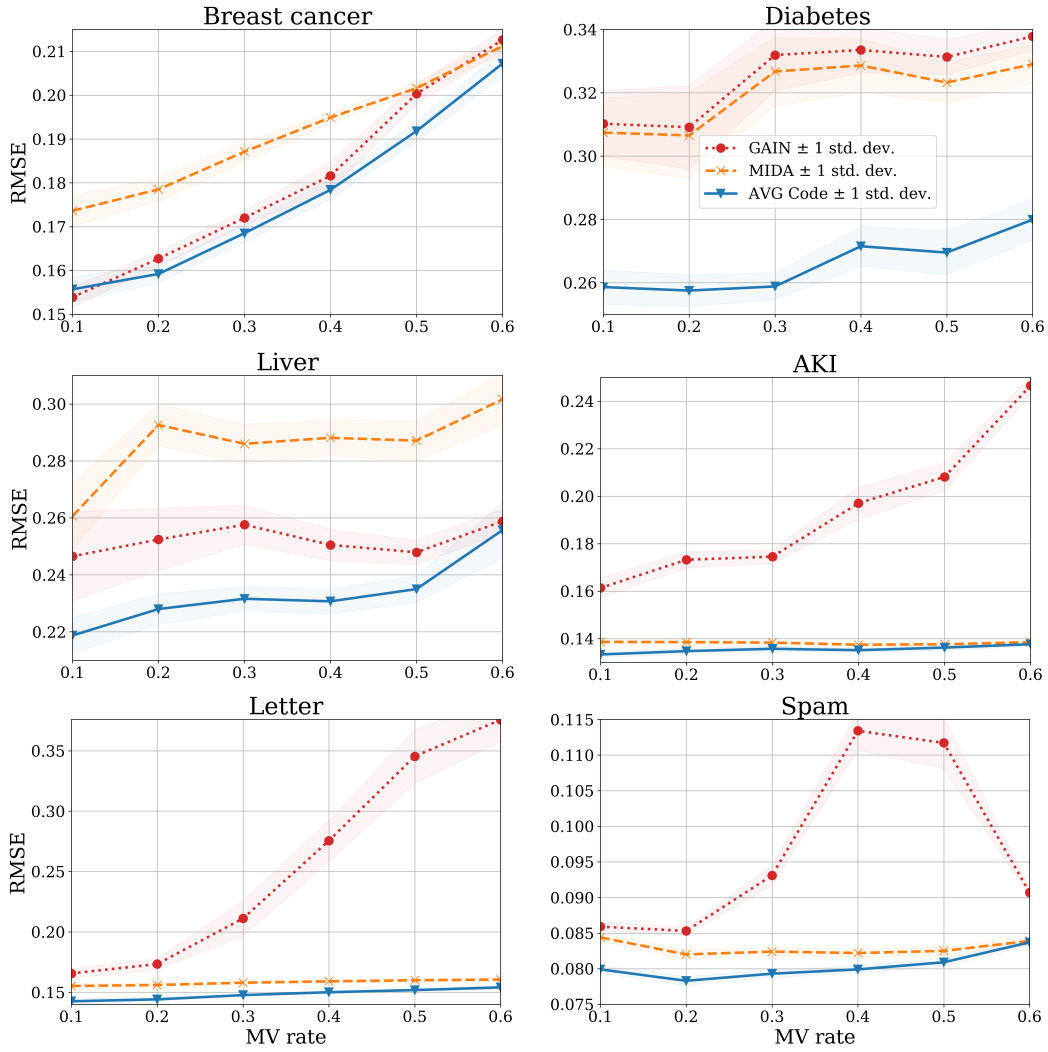


Fig. 4.4: Comparison of imputation mechanisms with missing complete at random missing values, varying the missing value rate for the available datasets.

MAR

For the appearance of MVs following the MAR mechanism, the proposed method is evaluated in a scenario more closely to a clinical environment. The appearance of MVs is not a random process. There is a dependency between attributes and MVs. According to the value of one or several attributes, MVs appear in other ones. The recommendations in [San+19] have been followed to emulate the generation of this type of MVs. In this case, the registers in the selected features, where MVs are synthetically generated, are deleted based on lower values of an observed feature. For this experiment, rates of

MVs vary from 10-60%, corresponding to a number of variables from 10-60% of the total of attributes for each dataset. To illustrate the experiment, let's consider the Spam dataset. The experiment starts with 10% of MVs in 6 attributes, then 20% of MVs in 12 attributes and so on.

In Fig. 4.5 it can be appreciated that the solution based on GANs has a volatile behavior in most of the datasets. However, it showed a better performance dealing with attributes with 10% of MVs. The proposed approach for 10% of MVs shows a weak performance for the datasets with few examples. For the rest of them, the AVG code solution has an overall better performance than MIDA and GANs. For the Cancer dataset, the MIDA solution has a better performance than the proposed approach. For the rest of the evaluations, the proposed approach showed better performance than GAN and compared with MIDA, for a range of MVs from 20-50% the proposed approach presented a better performance. Just for the datasets with more examples and 60% of MVs MIDA is competitive with the AVG code.

MNAR

To generate MVs following an MNAR mechanism, we followed the recommendations in [San+19]. In this case, the generation of the synthetic MVs is based on the variable itself. The recommendations suggest that the lower values of the variables are deleted. The rate of MVs vary from 10-60% and these MVs appears in 10-60% of the chosen attributes, as in MAR.

In Fig. 4.6 it can be appreciated that the AVG code shows a competitive performance with GANs for datasets with few examples and MV rates between 20-40%. The AVG code presents an overall better performance for the rest of the datasets, standing out among those datasets with more examples.

4.5 Discussion

In this work, a new alternative was proposed to impute MVs based on the integration of a MI in the latent spaces of an AE and the AVG code's computation. The AVG code combined the information from complex representations

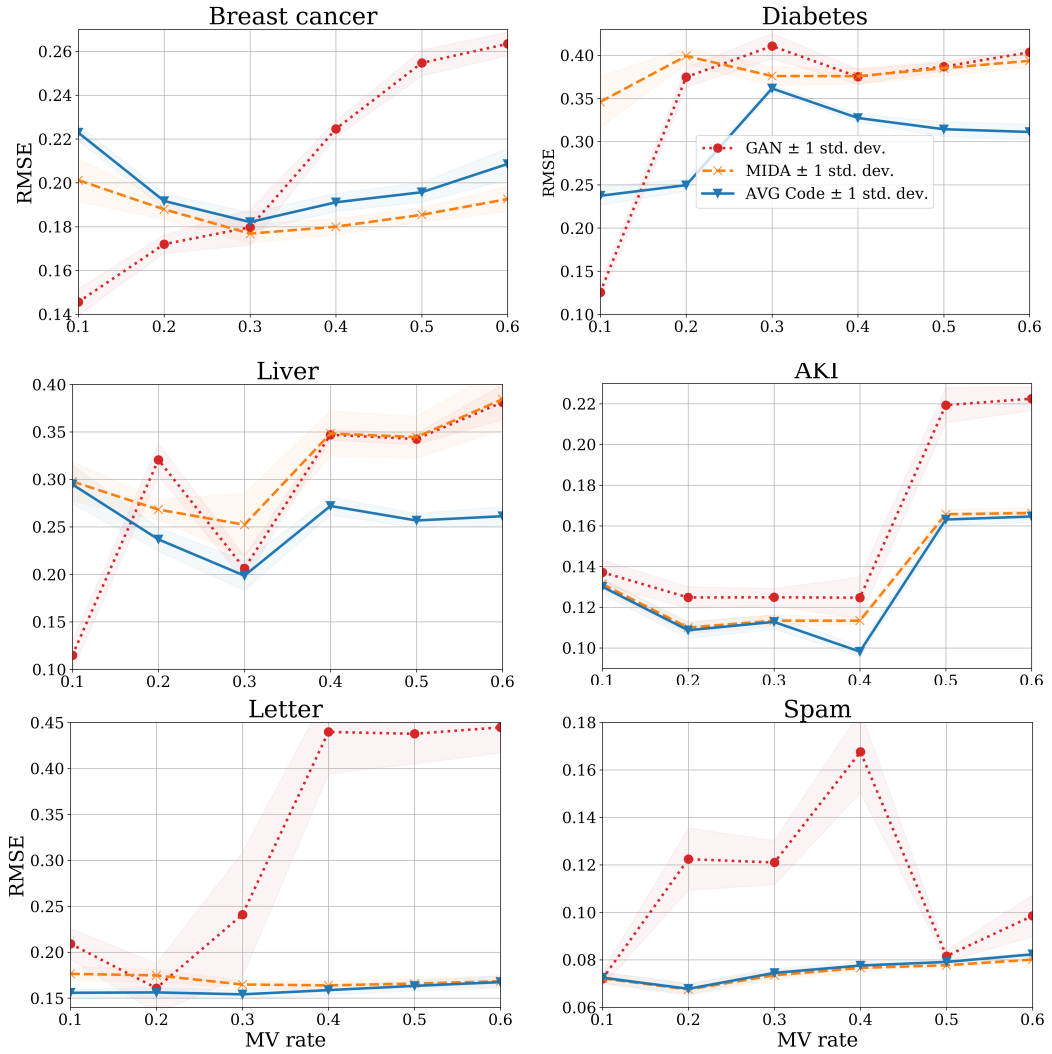


Fig. 4.5: Comparison of the imputation methods in missing at random, varying the missing value rate.

of data in latent spaces. The comparative results can be separated into two groups based on the dimension of the datasets. The first collection groups are the breast cancer, diabetes, and liver datasets, and the other three are grouped because they contain many more examples than the first group. Table. 4.2 shows the gain in RMSE the AVG code has over its competitors in all the performed experiments. This gain is computed based on the relation between the RMSE of the competitors and the AVG code, e.g. for GANs,

$$G_{GAN} = \frac{RMSE_{GAN}}{RMSE_{AVG}}. \tag{4.3}$$

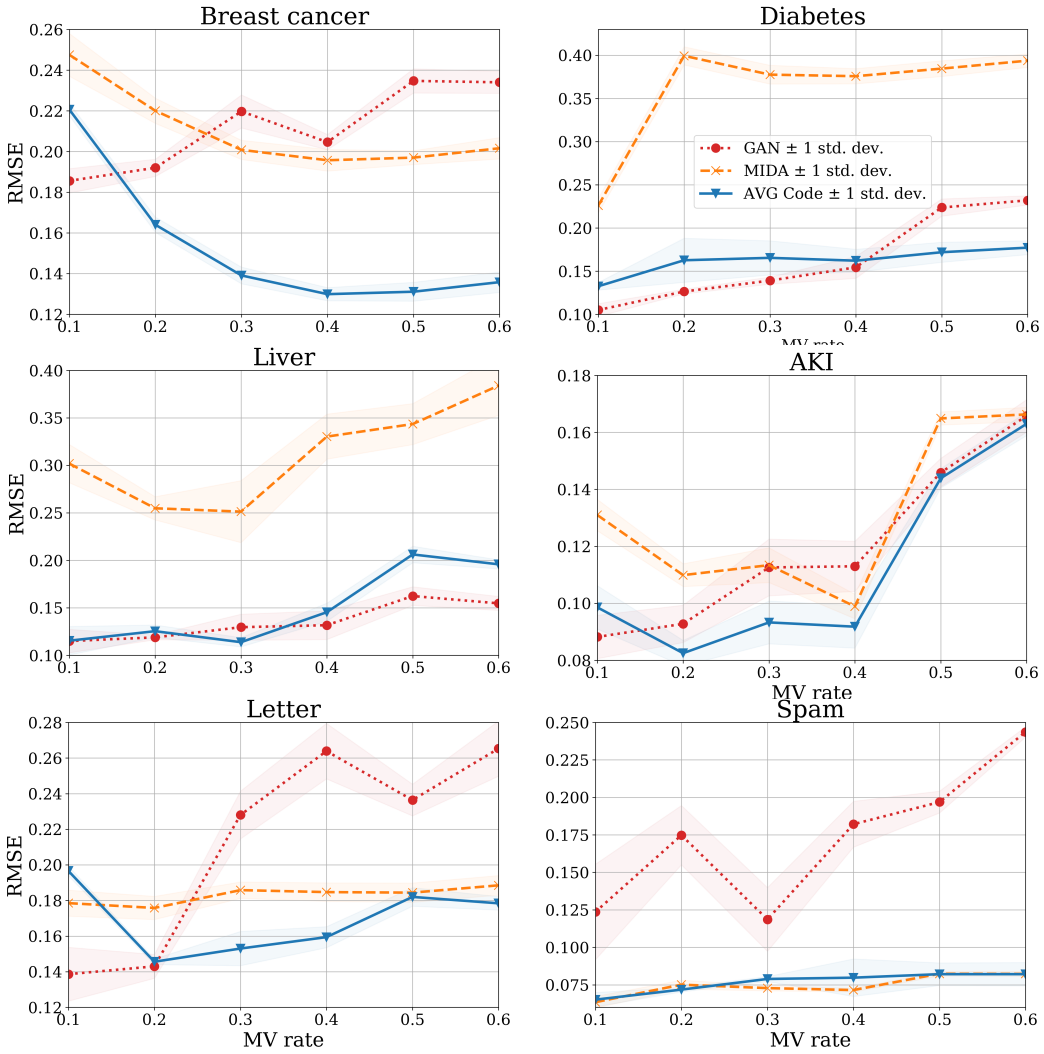


Fig. 4.6: Comparison of the imputation methods in missing not at random, varying the missing value rate.

For the experiment with MCAR values, the AVG code has an outstanding performance compared to its competitors. For the first collection of datasets, the AVG code offered a better reconstructive capacity in 97% of the evaluated cases. Compared with MIDA, the reconstructive error of the proposed method had a gain in 1.17 ± 0.07. Compared to GAN, the AVG code had an improvement of 1.12 ± 0.09. For the second group of datasets, the AVG code outperformed in all the evaluated cases compared with its competitors. With MIDA, the gain in RMSE was 1.04 ± 0.03. For GAN, the proposed method outperforms in 1.45 ± 0.40 in RMSE.

Tab. 4.2: Reconstructive gain using AVG codes compared to MIDA and GANs.

	MV rate	Cancer		Diabetes		Liver		Spam		Letter		AKI	
		GAN	MIDA	GAN	MIDA	GAN	MIDA	GAN	MIDA	GAN	MIDA	GAN	MIDA
MNAR	0.1	0.99	1.11	1.20	1.19	1.13	1.19	1.08	1.06	1.16	1.09	1.21	1.04
	0.2	1.02	1.12	1.20	1.19	1.11	1.28	1.09	1.05	1.20	1.08	1.29	1.03
	0.3	1.02	1.11	1.28	1.26	1.11	1.23	1.17	1.04	1.43	1.07	1.29	1.02
	0.4	1.02	1.09	1.23	1.21	1.09	1.25	1.42	1.03	1.84	1.06	1.46	1.02
	0.5	1.04	1.05	1.23	1.20	1.05	1.22	1.38	1.02	2.28	1.05	1.53	1.01
	0.6	1.03	1.02	1.21	1.18	1.01	1.18	1.08	1.00	2.44	1.04	1.79	1.01
MAR	0.1	0.65	0.90	0.53	1.46	0.39	1.01	0.99	1.00	1.34	1.13	1.05	1.01
	0.2	0.90	0.98	1.50	1.60	1.35	1.13	1.81	1.00	1.03	1.12	1.15	1.01
	0.3	0.99	0.97	1.14	1.04	1.04	1.27	1.62	0.99	1.56	1.07	1.11	1.01
	0.4	1.18	0.94	1.15	1.15	1.28	1.28	2.16	0.99	2.77	1.03	1.27	1.15
	0.5	1.30	0.95	1.23	1.22	1.33	1.34	1.03	0.98	2.68	1.02	1.34	1.02
	0.6	1.26	0.92	1.30	1.26	1.46	1.47	1.20	0.97	2.66	1.01	1.35	1.01
MNAR	0.1	0.84	1.12	0.79	1.70	1.00	2.62	1.90	0.97	0.70	0.91	0.89	1.33
	0.2	1.17	1.34	0.78	2.46	0.95	2.03	2.43	1.05	0.98	1.21	1.12	1.33
	0.3	1.58	1.44	0.84	2.28	1.14	2.21	1.50	0.92	1.49	1.21	1.21	1.22
	0.4	1.58	1.51	0.95	2.32	0.91	2.27	2.28	0.90	1.66	1.16	1.23	1.08
	0.5	1.79	1.50	1.30	2.24	0.79	1.67	2.40	1.00	1.30	1.01	1.01	1.15
	0.6	1.72	1.48	1.31	2.22	0.79	1.96	2.97	1.00	1.49	1.06	1.02	1,02

For the experiments with MAR values, for both groups of datasets, the proposed method outperformed in 69% and 81% of the experiments, respectively. The gain in reconstruction concerning GAN was 1.27 ± 0.13, while for MIDA, it was 1.27 ± 0.18. The proposed method is sensitive to low MVs ratios with datasets with few examples. This concern may be due to an under-fitting issue in the training of the AEs in both MIDA and AVG code. In the second group of datasets, the AVG code outperformed GAN 1.60 ± 0.61 and 1.27 ± 0.13 compared to MIDA. Although GAN presents better reconstruction when there are few MVs (10%) for datasets with few examples, the AVG code has a better reconstruction error when increasing the MV rate. For the second group of datasets and at most MV rates, the AVG code is robust enough to improve the performance of MIDA and, in a few cases, have similar performance.

In the last experiment, it was appreciated that the AVG code had similar behavior to MAR. In this case, the proposed method outperforms 72% and 81% of the experiments for the first and second groups of datasets, respectively. With this type of MVs, in the first group of datasets, it was possible to improve the reconstructive capacity with a gain of 1.45 ± 0.28 and 1.91 ± 0.44 for GAN and MIDA, respectively. The second group of datasets showed an improvement of 1.67 ± 0.60 and 1.13 ± 0.12 for GAN and MIDA, respectively. This MV mechanism showed to be more suitable for GANs than MIDA and competitive with the proposed method for the first group of datasets. For the

second one, the AVG code outperformed for most of the cases, being more representative for Spam and AKI.

4.6 Conclusions

In this work, the capacity to reconstruct missing information based on the computation of the AVG code from an AE was presented. The imputation capacity of a novel mechanism was evaluated that integrated a MI paradigm into the latent representation of information extracted by deep ANNs. The AVG code has a sufficiently robust imputation capacity to replace MVs to different MV rates and under several MV patterns, such as MCAR, MAR and MNAR. The AVG code demonstrated to maintain a low reconstruction error with different percentages of MVs. The variation used in the proposed approach, based on the training of an AE and the integration of MI in the latent space, revealed that the AVG code is adequate for high ratios of missing values, and the rest of the scenarios, its performance is not far from the best solutions. In conclusion, the work presented in this chapter is a solution with low algorithmic complexity that provides considerable benefits when reconstructing missing information. Finally, the integration of classical mechanisms such as MI to the latent spaces of a DL-based solution adds performance and robustness benefits compared to other literature methods based on deep learning.

5

Improving Mortality Predictive Models for Patients in ESRD: A Transfer Learning Approach

This chapter addresses the challenge of TL between healthcare entities to support learning tasks in those with low data volumes. Thus, two TL mechanisms are proposed in this chapter. They are based on sample and feature space augmentation. We start this chapter by introducing the topic of TL in medicine. Then we provide a background in the TL field, and finally, we present the proposed framework. The proposed framework is evaluated in predicting mortality in patients in ESRD, transferring information related to the mortality of patients with acute kidney injury from the massive database MIMIC-III. The proposed approach is compared with other TL mechanisms, showing an improvement of 6-11% in previous mortality predictive models. The integration of TL approaches into learning tasks in pathologies with data volume issues could encourage data-based medicine in a clinical setting.

The main contributions of this chapter are as follows:

- Explore the benefits of using a DL approach to TL in the clinical setting.
- Improve predictive models of mortality in ESRD patients by incorporating knowledge from a more extensive data set.
- Tackle the class imbalance issue through a solution based on TL.

- Propose a novel approach that uses the similarity of latent representations as a TL mechanism for feature augmentation.

5.1 Introduction

In the era of Big Data, DL is becoming a fundamental piece in the paradigm shift from evidence-based medicine to data-based medicine [Chi+18]. The increased availability of information, storage and processing capacity, and DL's capability to exploit complex relationships has allowed DL to significantly impact medical applications supported by Big Data [Pic+21]. Although the adoption of technologies that enable the collection of high volume of data in a clinical setting is growing, most medical centers do not have the infrastructure or the volume of patients to benefit from the learning capacity of DL [Yu+19]. Thus, integrating information from multiple health centers could significantly improve learning tasks in pathologies usually supported by a small volume of data. Implementing strategies for transferring data among domains could trigger DL solutions in a clinical setting and bring us closer to adopting data-based analysis for supporting clinical decisions.

The process of adapting and transferring knowledge among domains is known as TL [PY10]. The interest in TL in the medical field is increasing. In EHR, such as clinical images and biosignals, DL integration in a TL environment is proving to be an option that provides remarkable benefits [Lop+21; Lv+21; Cho+21; Shi+21]. Due to the capacity to exploit complex relationships that data may have and the data structures in such applications, e.g., spatial dependencies or time series, specialized ANN are commonly used. In such applications, the common TL approach pre-trains an ANN with data from one domain. Then the learned parameters are extracted for later use in applications in other domains [Maq+19; Byr+20; Mar+10]. However, this kind of approach is not suitable for data that contains heterogeneous structures because the procedure mentioned above constrains the input to be similar across domains. That is the case for applications that use other types of EHRs, like those that collect medical measurements of patients in a tabular way. Although there are solutions that incorporate such data into TL approaches [Des+17; EVM20], they use statistical analysis that does not

exploit complex relationships that the data may have. Thus, alternatives that include DL in TL solutions in pathologies with the type mentioned above data are still an open issue.

Transferring knowledge from high volume data sources to small datasets would allow DL to enhance learning tasks and be used to address class imbalance issues. This effect occurs because of the sudden changes in the patient's health condition. The volume of information generated for such events is smaller than the one associated with the rest of the follow-up. This effect commonly occurs in pathology prediction [Mac+19], rare event detection [Mac+18] or mortality prediction [Hai+17]. In Chapter 3, it was addressed the mortality prediction for patients in ESRD, however data imbalance was evidenced in the data due to minority of samples for the follow-up belong to the class deceased. There was a data imbalance in the range of 76-94%. Those issues cause low generalization of the learning models on the imbalanced samples, resulting in models whose performance is not acceptable for incorporation into clinical practice.

This chapter present a TL framework that uses information from a massive data source for supporting tasks in pathologies with a small data volume. The framework consists of two TL mechanisms used for sample and feature space augmentation in a target domain. AE are used to link both domains as a knowledge extraction mechanism. From AEs, codes are used as information bridges. For the sample increasing mechanism, they are used to create a feature mapping matrix used to transfer samples for a source domain to the target one. For the feature space augmentation, the TL mechanism is based on the computation of the average of the most similar codes of the target with the ones generated in the source domain. This TL framework is used to improve mortality prediction models in patients in ESRD. Volume and data imbalance issues are tackled with information extracted from patients with AKI from the massive database MIMIC-III. According to our knowledge, this is the first solution that integrates ANNs into a TL framework for solving learning tasks for kidney diseases.

5.2 Materials and methods

This section contains the necessary components to support the proposed TL framework. Classic AEs are the backbone of the knowledge extraction in the proposed framework. Moreover, two extensions of AEs widely used in the TL field are also addressed because they are used for performance comparison with the proposed method. Then, a method that has inspired part of the proposed framework is briefly explained. Finally, the problem that the methods can address is formally defined. Next, the necessary components to address the proposed TL framework are described.

5.2.1 Autoencoders

Thanks to the capabilities of an AE, it is possible to replicate the input at the output of the ANN and to obtain the latent spaces as described in Chapter 2. Thus, the encoder acts as a mapping function f_θ that transforms the input $\mathbf{x}$ into codes $\mathbf{h}$. Then codes are mapped back to reconstruction using the decoder function $g_{\theta'}$. Figure. 5.1 shows a recall of a simple AE, a deep one and their components.

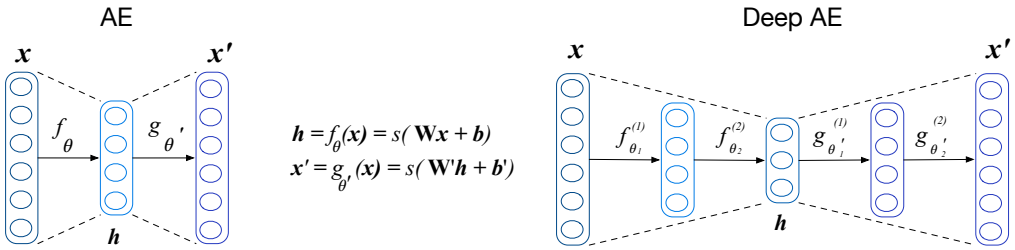


Fig. 5.1: Structure of single and multi layer AE.

Other alternatives that have shown outstanding performance using AEs in a TL environment are based on the application of stacked denoising AEs (SDA) [Vin+10] and its extension marginalized SDA (mSDA) [Che+12]. For the SDA, denoising AEs (DA) are trained. This type of AEs minimize the error between the input and a corrupted version, hence its name. To create the stack of DA, n DAs are trained. The first DA is trained with the corrupted version of the input, the second DA takes as input the code of the previous

DA, and so on, as is shown in the left side of Fig. 5.2. The training of each level follows the same process as a normal AE. At the end of the n trainings, the respective codes are used to create the final stacking that is shown in the right side of Fig. 5.2.

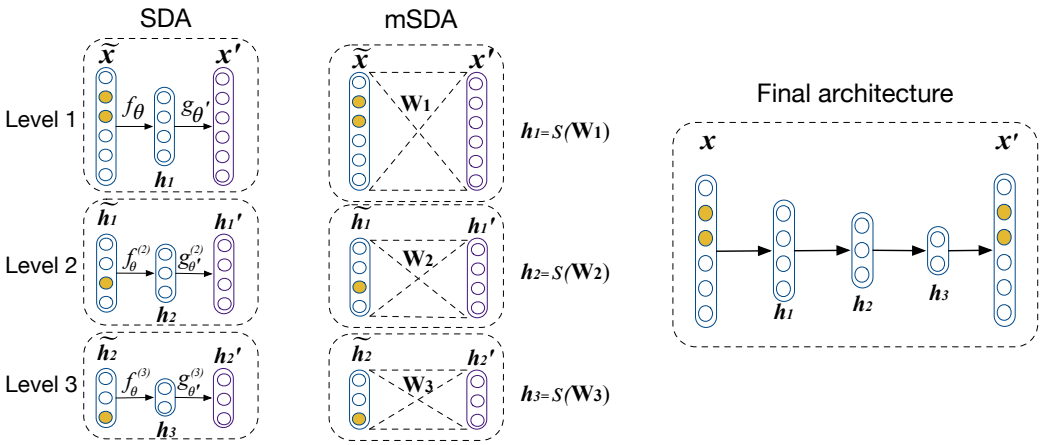


Fig. 5.2: SDA, mSDA and how their latent representations are used to create the final stacked.

On the other hand, the term marginalized in mSDA refers to the addition of noise to the inputs $\mathbf{x_i}$ in the iterations of the training process, e.g., different examples may be corrupted in every iteration. Thus, taking this into account, the cost function is transformed to

$$\mathcal{L} = \frac{1}{NM} \sum_{j=1}^{M} \sum_{i=1}^{N} \left\| \mathbf{x}_i - \mathbf{x}'_{i,j} \right\|^2, \tag{5.1}$$

where $\mathbf{x}'_{i,j}$ represents the j^{th} corrupted version of $\mathbf{x}_i$.

Then, with $\mathbf{X} = [\mathbf{x_1}, \ldots, \mathbf{x_n}] \in \mathbb{R}^{dxn}$, its m-times repeated versions $\overline{\mathbf{X}} = [\mathbf{X}, \ldots, \mathbf{X}]$ and its corrupted version $\widetilde{\mathbf{X}}$, Eq. 5.1 is reduced as

$$\mathcal{L} = \mathrm{tr}\left[\left(\overline{\mathbf{X}} - \mathbf{W}\widetilde{\mathbf{X}} \right)^{\top} \left(\overline{\mathbf{X}} - \mathbf{W}\widetilde{\mathbf{X}} \right) \right], \tag{5.2}$$

and its minimization solution can be expressed as,

$$\mathbf{W} = \mathbf{P}\mathbf{Q}^{-1} \text{ with } \mathbf{Q} = \widetilde{\mathbf{X}}\widetilde{\mathbf{X}}^{\top} \text{ and } \mathbf{P} = \overline{\mathbf{X}}\widetilde{\mathbf{X}}. \tag{5.3}$$

With a large m, i.e., $m \to \infty$, the bias estimation is reduced but the computational cost increases. To mitigate this issue, mSDA includes corruption probability p to a vector probability $\mathbf{q} = [1-p, \ldots 1-p, 1] \in \mathbb{R}^{d+1}$. $\mathbf{q}_i$ represents the probability of a feature i surviving the corruption. Thus the expectation for Eq. 5.3 can be computed and $\mathbf{W}$ can be expressed as follows,

$$\mathbf{W} = \mathbb{E}[\mathbf{P}]\,\mathbb{E}[\mathbf{Q}]^{-1} \text{ with } \mathbb{E}[\mathbf{P}]_{i,j} = \mathbf{S}_{i,j}\mathbf{q}_j,\ \mathbf{S} = \mathbf{X}\mathbf{X}^\top \text{ and,} \tag{5.4}$$

$$\mathbb{E}[\mathbf{Q}]_{i,j} = \begin{cases} \mathbf{S}_{i,j}\mathbf{q}_j\mathbf{q}_j & if i \neq j \\ \mathbf{S}_{i,j}\mathbf{q}_i & otherwise. \end{cases} \tag{5.5}$$

With $\mathbf{W}$, nonlinear function s is applied, then nonlinear features can be extracted as $\mathbf{h} = s(\mathbf{W}\mathbf{x})$. Such nonlinear functions may include tangent hyperbolic (tanh), sigmoid or Rectified Linear Unit (ReLU).

5.2.2 Hybrid heterogeneous transfer learning

The so-called Hybrid Heterogeneous Transfer Learning (HHTL) proposed in [ZPT19] is a TL framework for transferring knowledge between two heterogeneous domains using mSDAs. HHTL solves a learning task related to labelling samples from one domain using information from the other one. The target domain is defined as $\mathbf{D}_T = \{(\mathbf{x}_{T_i}, \mathbf{y}_{T_i})\}_{i=1}^{n_2}$ and the source domain as $\mathbf{D}_S = \{\mathbf{x}_{S_i}\}_{i=1}^{n_1}$, where $\mathbf{x}_{S_i} \in \mathbb{R}^{d_S x 1}$ and $\mathbf{x}_{T_i} \in \mathbb{R}^{d_T x 1}$ are the data and $\mathbf{y}_{\mathbf{T_i}}$ labels; n_1 and n_2 the total of samples and d_S and d_T their features. The information to be transferred is the hidden representations extracted from mSDAs for each domain. mSDAs are trained in both domains with k ($k = 1, \ldots, K$) hidden layers as is illustrated in Fig. 5.3. Then, latent representations $\mathbf{H}_{S,1}, \ldots, \mathbf{H}_{S,k}$ and $\mathbf{H}_{T,1}, \ldots, \mathbf{H}_{T,k}$ are extracted and then related through mapping matrices, $\mathbf{G}_k$, as is shown in Fig. 5.3. These matrices acts as TL bridges for the hidden representations in both domains. To find each $\mathbf{G}_k$, they minimize the objective,

$$\min_{\mathbf{G}_k} \|\mathbf{H}_{S,k} - \mathbf{G}_k\mathbf{H}_{T,k}\|^2 + \lambda \|\mathbf{G}_k\|^2 . \tag{5.6}$$

Once $\mathbf{G}_k$ is computed, new samples $\mathbf{X}_S^*$ along with its hidden representations $\mathbf{H}_{S,k}^*$ can be transferred to $\mathbf{D}_T$, i.e., $\mathbf{H}_{S\rightarrow T,k}^* = \mathbf{G}_k\mathbf{H}_{S,k}^*$. $S \rightarrow T$ refers to the transfer from $\mathbf{D}_S$ to $\mathbf{D}_T$. Then, to solve the learning task, they create a new feature space with the hidden representations of $\mathbf{D}_T$, i.e., $\mathbf{Z}_T = \left[\mathbf{H}_{T,1}^\top \dots \mathbf{H}_{T,k}^\top\right]^\top$. Then, a classifier $\{(\mathbf{Z}_T, \mathbf{y}_T)\}$ is trained. With the latent transferred representations, a similar feature space $\mathbf{Z}_{S\rightarrow T} = \left[\left(\mathbf{H}_{S\rightarrow T,1}^*\right)^\top \dots \left(\mathbf{H}_{S\rightarrow T,k}^*\right)^\top\right]^\top$ is created. Finally, with the trained classifier they predict over $\mathbf{Z}_{S\rightarrow T}$ the labels for $\mathbf{D}_S$ samples.

Part of the sample augmentation for the proposed approach is based on the computation of $\mathbf{G}_k$, with the difference that we only use it to relate the codes of the AEs and not the rest of the latent representations of each hidden layer. Hence we compute a single $\mathbf{G}$.

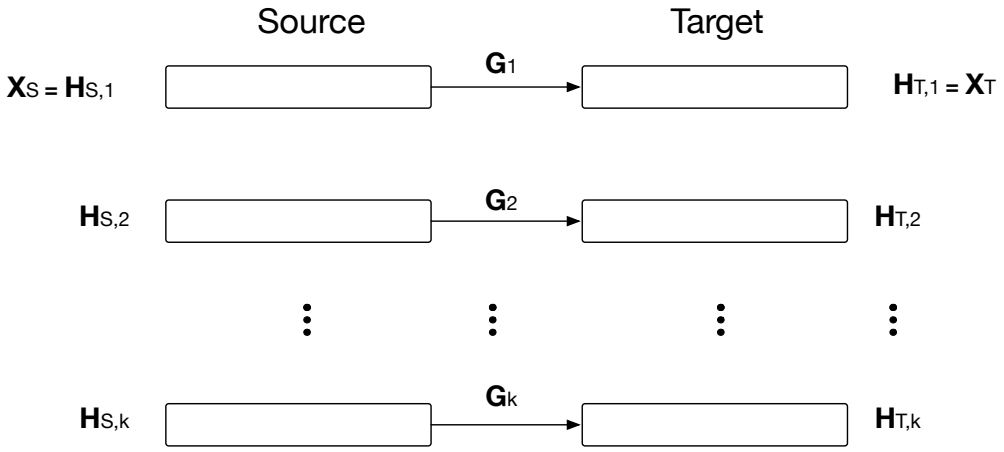


Fig. 5.3: HHTL for transferring hidden representations, $\mathbf{H}$, between source and target domain. $\mathbf{H}$ are extracted from trained mSDAs.

5.2.3 Problem definition

Given a set of labelled data from the source and target domains, $\mathbf{D}_S = \{(\mathbf{x}_{S_i}, \mathbf{y}_{\mathbf{S_i}})\}_{i=1}^{n_1}$ and $\mathbf{D}_T = \{(\mathbf{x}_{T_i}, \mathbf{y}_{\mathbf{T_i}})\}_{i=1}^{n_2}$, respectively, where $\mathbf{x}_{S_i} \in \mathbb{R}^{d_S x 1}$ and $\mathbf{x}_{T_i} \in \mathbb{R}^{d_T x 1}$ are the data and $\mathbf{y}_{\mathbf{S_i}}$ and $\mathbf{y}_{\mathbf{T_i}}$ their labels; n_1 and n_2 the total of samples and d_S and d_T their features. The aim of TL in this chapter is to improve the learning task in $\mathbf{D}_T$ with information from $\mathbf{D}_S$. The transfer

of knowledge is carried out by managing codes of trained AEs from both domains in two manners. The first on follow the next steps:

- Transfer samples from one domain to another through the computation of a feature mapping matrix $\mathbf{G}$, as in HHTL.
- Maps codes from one domain to the other one using $\mathbf{G}$.
- Transfer a sample $\mathbf{x}_S^*$ to $\mathbf{D}_T$ through $\mathbf{G}\mathbf{h}_S^*$, where $\mathbf{h}_S^*$ is the code of $\mathbf{x}_S^*$.

The second mechanism attempts to increase the feature space of $\mathbf{D}_T$ with the average of the most similar codes, computed by a similarity metric, the euclidean distance between the codes, that compare each code from $\mathbf{D}_T$ with the entire set of codes from $\mathbf{D}_S$. The increase in samples and features may reinforce the learning task in $\mathbf{D}_T$. Fig. 5.4 shows a scheme of the mechanisms that are used to enhance the learning task in $\mathbf{D}_T$.

5.3 Proposed method

The proposed approach is motivated by the availability of massive sources of medical data and the potential benefits of integrating them to encourage the adoption of data-based medicine. This integration makes it possible to exploit the learning capacity that DL has on massive data. Thus, two TL mechanisms are proposed to enhance the performance in learning tasks in a clinical environment. Specifically, the predictive capacity of mortality predictors for patients in ESRD will be evaluated using a TL framework. It is proposed to apply TL approaches to increase both samples and feature space in $\mathbf{D}_T$ using information from $\mathbf{D}_S$. As mentioned previously, such mechanisms may tackle class imbalance issues to improve the predictive capacity of the previous mortality models in ESRD.

In the proposed framework, both domains contain labeled samples. AEs are used to extract data representation into their codes for sample and feature space augmentation. Thus, the framework relies on two main components:

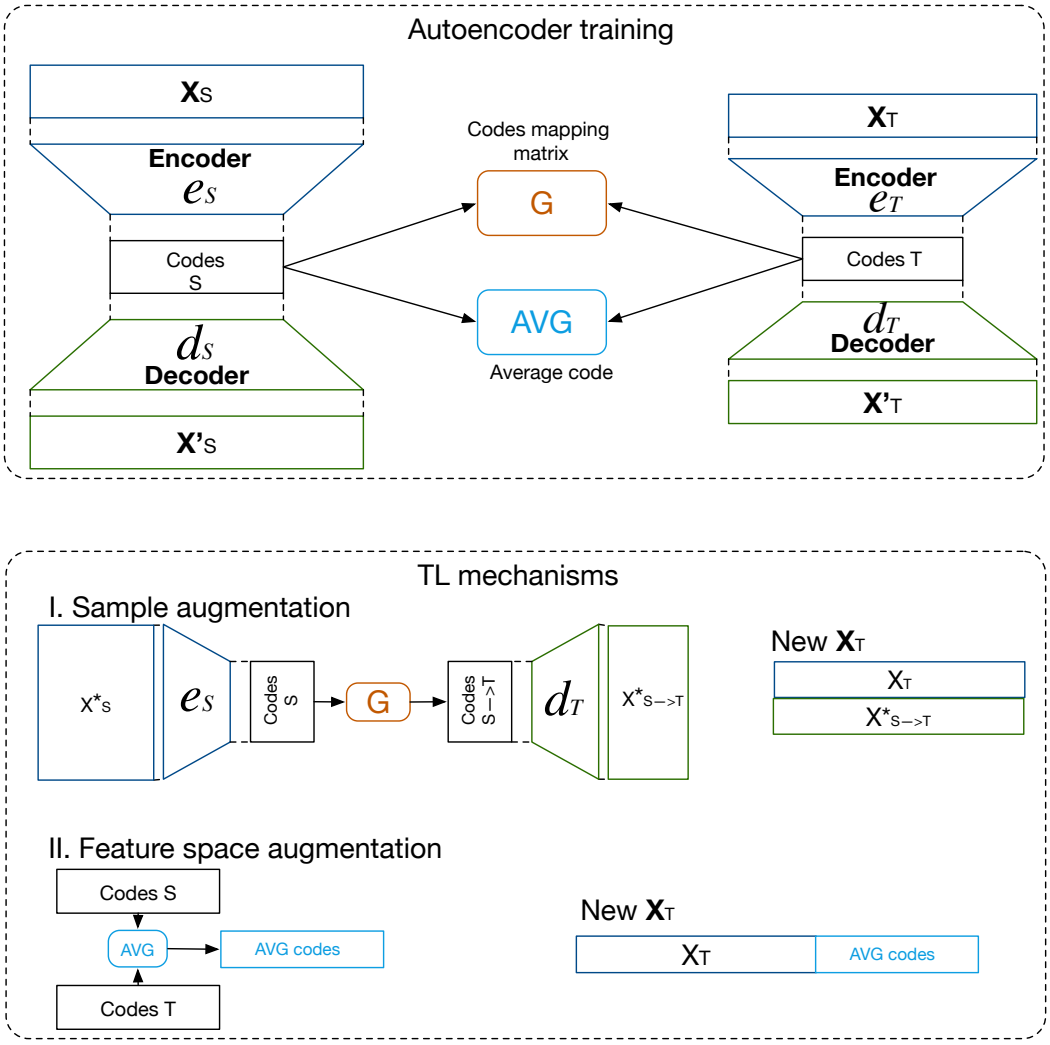


Fig. 5.4: Scheme of proposed method for transfer of samples between domains and feature space augmentation to support learning tasks in the target domain. HHTLM refers to the modified version of Hybrid Heterogeneous Transfer Learning and TLVA to transfer learning based on average codes.

- Sample augmentation using a mapping matrix $\mathbf{G}$, encoder and decoder functions in both domains to transfer and reconstruct codes from $\mathbf{D}_S$ in $\mathbf{D}_T$.
- Feature space augmentation based on the computation of the average of the most similar codes.

5.3.1 Sample augmentation-TLCO

For augmenting samples in $\mathbf{D}_T$, a three-stage TL mechanism is used. Initially, from both domains, AEs are trained, and the codes are extracted to compute a mapping matrix $\mathbf{G}$, as in HHTL. It is worth mentioning that, unlike HHTL, in our approach, we reinforce knowledge transfer by considering the reconstruction of the codes of one domain using the decoders of the other domain. We refer to this method as TL by codes or TLCO. In a second stage, $\mathbf{G}$ is used to transfer codes from $\mathbf{D}_S$. Thus, $\mathbf{H}_S^*$, produced by data $\mathbf{X}_S^*$ in $\mathbf{D}_S$ are first transferred to $\mathbf{D}_T$. Then, the decoder function in $\mathbf{D}_T$ reconstructs the transferred codes in such a domain. The parameters of the decoder function of trained AEs in each domain allow the reconstruction of their codes. The decoders in the opposite domains and the mapping matrix between the codes can be used as a reinforcement mechanism for cross-domain knowledge transfer. Once the samples are reconstructed, they are used to increase $\mathbf{D}_T$. This last step allows for tackling the class-imbalance issue. Fig. 5.5 illustrates how this TL mechanism is carried out using datasets from kidney diseases. It is also provided the detailed steps of the proposed method in Algorithm 2.

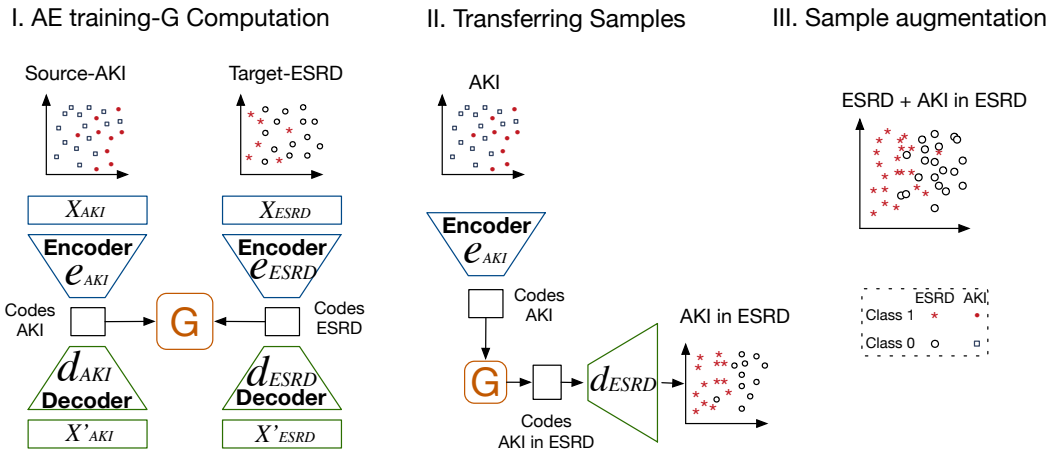


Fig. 5.5: Scheme of proposed method for transfer of samples between domains and the support of a learning task in the target domain using TLCO.

5.3.2 Feature space augmentation-TLAV

For feature space augmentation, the TL mechanism is based on the computation of averaging the most similar codes from $\mathbf{D}_S$ to codes in $\mathbf{D}_T$. We refer

Algorithm 2 Increasing samples using TLCO

Input: Data from both domains, $\lambda = 0.001$: $\mathbf{D}_S = \{(\mathbf{x}_{S_i}, y_{S_i})\}_{i=1}^{n_1}$, $\mathbf{D}_T = \{(\mathbf{x}_{T_i}, y_{T_i})\}_{i=1}^{n_2}$

3 Train AEs with $\mathbf{X}_S$ and $\mathbf{X}_T$. Extract encoder (e) and decoder (d) functions from both domains, and the latent representations-codes $\mathbf{Z}$: $\mathbf{Z}_S = e_S(\mathbf{X}_S)$,

$\mathbf{X}'_S = d_S(\mathbf{Z}_S)$
$\mathbf{Z}_T = e_T(\mathbf{X}_T)$, $\mathbf{X}'_T = d_T(\mathbf{Z}_T)$;

4 Learn heterogeneous feature mapping $\mathbf{G}$:

$$\min_{\mathbf{G}} \|\mathbf{Z}_S - \mathbf{G}\mathbf{Z}_T\|^2 + \lambda \|\mathbf{G}\|^2;$$

5 Augment samples in $\mathbf{D}_T$ with samples from $\mathbf{D}_S$:

$\mathbf{X}^*_{S\to T} = \mathbf{G}^\top \mathbf{X}^*_S$
$\mathbf{X}^*_T = [\mathbf{X}_T \quad \mathbf{X}^*_{S\to T}]$, $\mathbf{y}^*_T = [\mathbf{y}_T \quad \mathbf{y}_S]$

Note: $S \to T$ refers to the transfer from $\mathbf{D}_S$ to $\mathbf{D}_T$.

6 Train a classifier f with $\{(\mathbf{X}^*_T, \mathbf{y}^*_T)\}$

Output: Classifier f

as average code or AVG_{codes}. They increase features for every sample in $\mathbf{D}_T$. We refer to this approach as TL by AVG_{codes} or TLAV. As the information that best represents the data after AEs' training is encapsulated in their codes, this approach uses the AVG_{codes} as extra features that may enhance the predictive capacity of learning models.

The proposed method is summarized into three stages (see Fig. 5.6). Initially, AEs are trained in both domains, and their codes are compared. For TLAV, it is hypothesized that similar codes represent similar information even from different domains. Thus, every code from $\mathbf{D}_T$ is compared with all the codes from $\mathbf{D}_S$. The Euclidean distance is computed as a similarity metric for the comparison (see Eq. 5.7). Then, the most similar codes are filtered based on a similarity threshold, ϵ, that indicates the percentage of the most similar codes. Based on ϵ, sets of n_3 (as in Fig. 5.6) codes from $\mathbf{D}_S$ are extracted for each code in $\mathbf{D}_T$. Then, in the second stage, the codes' sets are summarized in their average to find a more robust representation. Finally, the AVG_{codes} are merged, then concatenated to the samples in $\mathbf{D}_T$ and finally, such new feature space is used to perform the learning task in $\mathbf{D}_T$.

$$d\left(\mathbf{h_S}, \mathbf{h_T}\right) = \sqrt{\sum_{i=1}^{n} \left(h_{Si} - h_{Ti}\right)^2}. \tag{5.7}$$

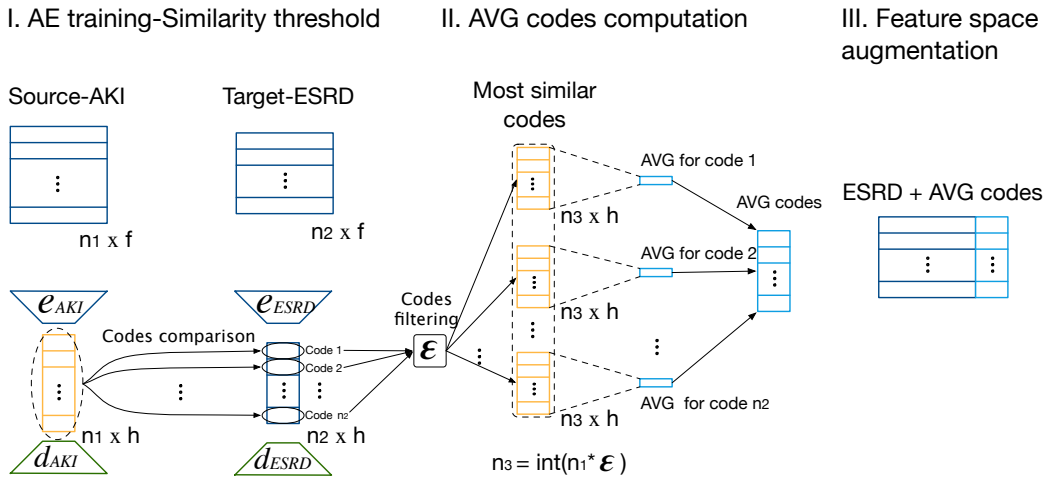


Fig. 5.6: Scheme of proposed method for transferring AVG_{codes} from $\mathbf{D}_S$ to $\mathbf{D}_T$ using TLAV.

5.4 Experimental setup

In this section, all the necessary components and the evaluation of the proposed TL framework in predictive mortality models for patients in ESRD are presented. Initially we describe the datasets where the TL mechanisms are evaluated. Then, in order to compare the benefits of the proposed framework, a modified version of HHTL is used as competitor for sample augmentation and feature space augmentation. At the end of the section, the experiments performed and their respective evaluations are presented.

5.4.1 Datasets

For this work, they are used two datasets related to kidney disease. The learning tasks in both datasets is related to mortality prediction, one for the follow-up of patients in ESRD and the another one for patients with AKI in ICU. The objective is to improve mortality predictive models for patients in ESRD with data from patients with AKI. Next, they are described in detail.

ESRD: Information for $\mathbf{D}_T$ is part of a previous study for predicting mortality in ESRD patients [Mac+20]. These data were taken from the Information System of the Parc Tauli University Hospital, from the Haemodialysis (HD) Unit at the Nephrology Department from 2007 to 2018. Data transfer passed through the ethics committee (Code 2018/508) and subsequently anonymised following the usual protocol. In such study, information from the follow-up of 261 patients in ESRD from the beginning of the haemodialysis treatment until the deceased event, were collected. The feature space include a mixture of categorical and continuous measurements from laboratory test, diagnoses and variables measured during the haemodialysis sessions. In total, there are 53 features. During their follow-up, such patients have generated 8229 samples. Four datasets were generated based on the date of death of the patients, hence the mortality models have labels associated to 1, 2, 3 and 6 months before the death event.

AKI: The dataset for $\mathbf{D}_S$ has been extracted from MIMIC-III database [Joh+16b]. Such massive database contains information from more than 40000 patients in ICU. From MIMIC-III, patients with AKI were filtered based on the kidney disease improving global outcomes (KDIGO) clinical practice guideline [Ink+14]. Information from 4152 patients with 31 features were extracted. The total of samples in such cohort contains more than 125000 samples. Their follow-up includes demographics, diagnoses, laboratory tests, physiological measurements during the ICU stay and the in-hospital mortality label.

5.4.2 Experimental results

In evaluating the predictive capacity of the TL mechanisms in the mortality models for patients in ESRD, several experiments are defined based on the way of transferring knowledge. As AEs are the backbone of the proposed TL framework, we initially compared the performance of a deep AE with an mSDA applying TLCO and TLVA. Then, we compare the methods with HHTL. HHTL is modified in this work for sample and feature space augmentation. Next, the setup mSDA and HHTL is listed:

- Deep AE vs. mSDA: we designed a baseline to choose which type of AE better suits the data. We train deep AEs with two hidden layers for the encoder and decoder functions. Then a two-level mSDA is trained. The codes are extracted from the deep AE to perform TLCO and TLAV. For mSDA, the hidden representations from the second level are extracted as codes, and TLCO and TLAV are applied to them.

- HHTL: HHTL has been widely compared with other approaches in the TL literature, showing a better performance than its competitors [ZPT19]. The modified versions of HHTL, for sample and feature space augmentation are based on the management of the hidden representations for the levels of the trained mSDAs. As stated in 5.2.2, such hidden representations are extracted from hidden layers to create new feature spaces,

$$\mathbf{Z}_T = \left[\mathbf{H}_{T,1}^\top \dots \mathbf{H}_{T,K}^\top\right]^\top, \text{and } \mathbf{Z}_{S\to T} = \left[\left(\mathbf{H}_{S\to T,1}^*\right)^\top \dots \left(\mathbf{H}_{S\to T,k}^*\right)^\top\right]^\top, \tag{5.8}$$

  then sample augmentation is carried out adding samples from AKI to ESRD in their respective new feature space, i.e., $\mathbf{Z}_{samples} = [\mathbf{Z}_T;\ \mathbf{Z}_{S\to T}]$. As HHTL is a method to transfer samples, for feature space augmentation, as in the proposed approach, we use averages of most similar hidden representations from $\mathbf{Z}_{S\to T}$ to augment $\mathbf{Z}_T$, i.e., $\mathbf{Z}_{features} = [\mathbf{Z}_T\ \ \mathbf{AVG}_{\mathbf{Z}_{S\to T}}]$.

The performance of the experiments is evaluated on the learning task in ESRD. The AUROC is used as a metric to find the best models in the experiments. Recalling that AUROC relates the sensitivity and specificity of a classifier. Its values lie between 0 and 1, with 1 being the perfect classifier and 0.5 as a random one. The baseline performance and classifiers used in this work are based on long short-term memory ANNs, used in Chapter 3 for every short term mortality horizons. All the reported experiments used 5-folds for cross-validation. Two sets of experiments have been defined to determine the performance of the proposed methods.

TLCO-Sample augmentation

In ESRD data, the class imbalance varies according to the mortality horizon. Information on how the sample labels are computed can be found in the previous study [Mac+20]. Table. 5.1 shows the class imbalance caused by each mortality horizon. To implement TLCO, initially, AEs with two hidden layers are trained for both datasets. Then, their codes are extracted. The hyperbolic tangent (Tanh) activation function is used for the hidden layers and the Sigmoid for the output layer in AKI. For the ESRD dataset, rectified linear unit (ReLU) activation function for hidden layers and Sigmoid at the output layer were used. Dropout of 0.1 and batch normalization were applied in the hidden layers of the AEs to avoid overfitting. Once the AEs are trained, the mapping matrix $\mathbf{G}$ is generated using codes from both domains. Then, the codes from AKI are transferred to the latent space of the ESRD domain using $\mathbf{G}$. Finally, transformation is reconstructed using the decoding function of the trained AE in ESRD. For mSDA, a Tanh was used as a non-linear function to compute the codes. Next, three experiments are listed to find the best performance for the mortality predictors.

Tab. 5.1: Imbalance of samples for the prediction of mortality in patients in ESRD. **Class 0** and **Class 1** refer to samples in alive and deceased classes, respectively.

Mortality	Class 0	Class 1	Imbalance (%)
1	7734	495	93.6
2	7488	741	90.1
3	7251	978	86.5
6	6632	1597	75.9

- Code dimension: the dimensions of codes in both domains are evaluated to find a high-level representation of the data that allows us to transfer valuable information. Thus, the combination of dimensions that presents the best overall performance for the prediction task is empirically found. In Fig. 5.7 (a), it is denoted the dimension of the codes for the deep AE in AKI and ESRD as S_*, and T_*, where $*$ refers to the dimensions of the code, e.g., S_30 T_40 refers to the combination of having trained AEs with codes of dimension 30 and 40 for AKI and ESRD, respectively. It is also shown the performance of mSDA. Moreover, it should be noted that in mSDA, the dimension of the codes

has the same input data dimension, which is why only one predictor is observed for mSDA in the figure. Also, it can be appreciated that most of the combinations present a higher performance than the baseline one. Although mSDA outperforms better than most predictors, the deep AE with 30 and 80 codes in AKI and ESRD offers a better predictive capacity than mSDA.

- Sample augmentation in ESRD: this experiment evaluates how the increase of samples in the training set affects the predictive models of mortality in ESRD. For this experiment, three possible scenarios were defined. Initially, the data imbalance in ESRD is intentionally increased. Thus, only Class 0 in AKI samples are transferred to the ESRD training set. This transfer is carried out to evaluate whether an adverse effect is linked to the increase in data imbalance. In the second scenario, the training set samples are increased, but only those that belong to AKI Class 1 are transferred. In this case, the aim is to balance the imbalanced class. Finally, in a third scenario, both classes are transferred from AKI to ESRD. Therefore, we evaluate both the effect of the increase in samples and the reduction of data imbalance in the predictive models. Table. 5.2 shows how the data imbalance varies for each scenario. In Fig. 5.7 (b), it can be appreciated that increasing samples in the training set of the ESRD data does not imply, in most of the scenarios, a deterioration in the predictive models. On the other hand, when the number of samples increases, the learning models present a better predictive capacity considering the imbalance ratio.

- Comparing with HHTL: to evaluate the performance of HHTL, the number of transferred samples was adjusted following the third scenario in the previous experiment. Thus, in Fig. 5.7 (c) it can be appreciated that although HHTL for upsampling or HHTL4S improves the base predictive models, it has a lower performance than that found by deep AE.

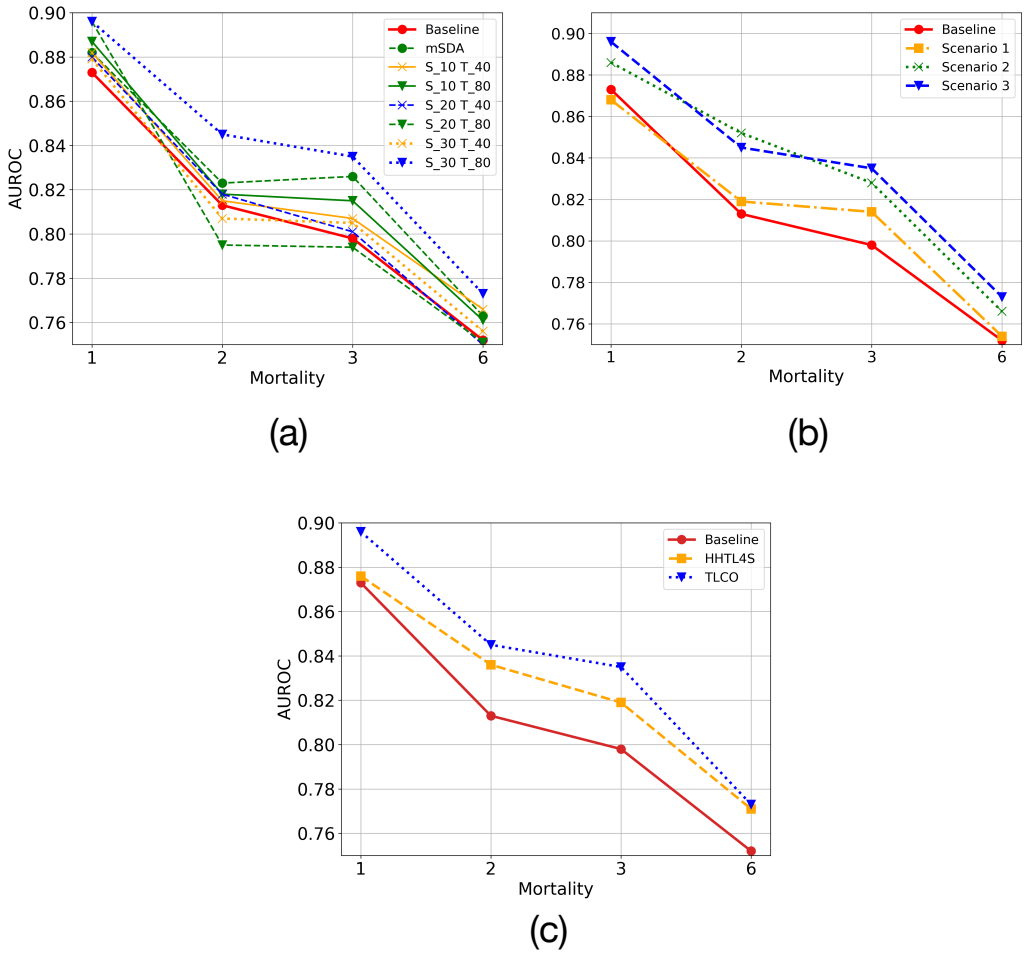


Fig. 5.7: Comparison results transferring samples from AKI to ESRD using TLCO. (a) shows the performance of mSDA and TLCO modifying dimension of codes in source (S) and target (T) domains. (b) shows three possible scenarios to tackle data imbalance in ESRD. (c) compare proposed solution with HHTL.

Tab. 5.2: Imbalance in ESRD generated by increasing training samples in ESRD from AKI. Scenario 1, 2 and 3 refer to the transfer of samples from the Class 0, 1 and combining both classes, respectively.

2*Mortality	Generated data imbalance (%)		
	Scenario 1	**Scenario 2**	**Scenario 3**
1	95.2	73.4	80.0
2	92.6	69.2	77.1
3	90.1	74.9	74.2
6	82.7	52.3	65.7

TLAV-Feature space augmentation

In the performance evaluation of TLAV, the same hyperparameters for training AEs as in TLCO are used for TLAV and its competitors. With TLAV, the augmented feature space is based on the computation of AVG_{codes}. As a recall, such AVG_{codes} are computed based on the comparison of each code in ESRD with all the codes in AKI. Each comparison generates a set of codes that are filtered by a similarity threshold (ϵ) and summarized into AVG_{codes}. Parametric analysis and comparison with mSDA configuration and HHTL for feature augmentation (HHTL4F) are carried out. Thus, three experiments were carried out to find the best models to enhance the learning task in ESRD. They are explained below.

- Code dimension: the first parameter that controls the behavior of TLAV is the dimension of the codes (dim_h). This parameter reflects the ability of AEs to represent information in latent spaces under the TL methodology of TLAV. In this experiment, ϵ is set to 0.4. Fig. 5.8(a) shows scenarios where the input information is compressed or dispersed according to the value of dim_h. It can be appreciated that bottleneck type deep AEs offer better overall performance than sparse type deep AEs. The best solution is the one with $dim\_h = 10$.
- Tuning similarity threshold (ϵ): with Euclidean distances from ESRD and AKI codes, a proportion of these codes is chosen using ϵ. ϵ controls the amount of more similar AKI codes used to compute the average one. Once every set of codes from AKI are extracted, their AVG_{codes} are computed and used to increase the feature space for each ESRD sample. Table. 5.3 shows the performance of the predictive models varying ϵ. It can be appreciated that increasing the number of codes for the computation of their average reflects a slight improvement in the predictive models. However, from an ϵ of 0.3 or 0.4, more codes do not imply a considerable increase in the predictive models. Compared with its competitors, TLAV based on deep AEs presents a better performance when more codes are included for the average computation. Using the three methods, taking 40% of the most similar AKI codes for each ESRD code presents the most balanced performance for mortality prediction.

Tab. 5.3: Comparison results applying TLAV with mSDA and HHTL4F using AVG_{codes} concept and varying similarity threshold ϵ. In bold the best predictive models for each mortality horizon.

		Mortality			
ϵ	**TL method**	**1**	**2**	**3**	**6**
0.01	mSDA	0.857	0.839	0.816	0.761
	HHTL4F	0.878	0.824	0.820	0.757
	TLAV	0.887	0.849	0.816	0.758
0.1	mSDA	0.856	0.840	0.809	0.761
	HHTL4F	0.879	0.831	0.818	0.759
	TLAV	0.891	0.854	0.816	0.763
0.2	mSDA	0.859	0.834	0.811	0.760
	HHTL4F	0.891	0.834	0.819	0.758
	TLAV	0.901	0.857	0.820	0.761
0.3	mSDA	0.863	0.841	0.820	0.760
	HHTL4F	0.895	0.837	0.822	0.758
	TLAV	0.906	0.860	**0.823**	**0.765**
0.4	mSDA	0.877	0.842	0.819	0.758
	HHTL4F	0.894	0.835	0.821	0.760
	TLAV	**0.909**	**0.862**	**0.823**	0.763
0.5	mSDA	0.875	0.842	0.818	0.759
	HHTL4F	0.891	0.836	0.819	0.759
	TLAV	0.904	0.861	0.821	0.765

TLAV with deep AEs is the best option to increase the feature space in ESRD.

TLAV-HHTLM

In the last experiment, TLAV is combined with TLCO. Such combination is performed in a cascade way. The parameters that control TLCO and TLVA are found in previous experiments. Thus, in the first stage, the ESRD feature space increases using TLAV. Then, TLCO is applied to this new version of ESRD to increase the number of samples. Table. 5.4 presents the performance of the combination, compared to the literature methods and the best predictors by TLAV and TLCO separately. It can be seen that the combination of the two proposed methods has a considerable influence on the performance of the predictive models for short-term mortality.

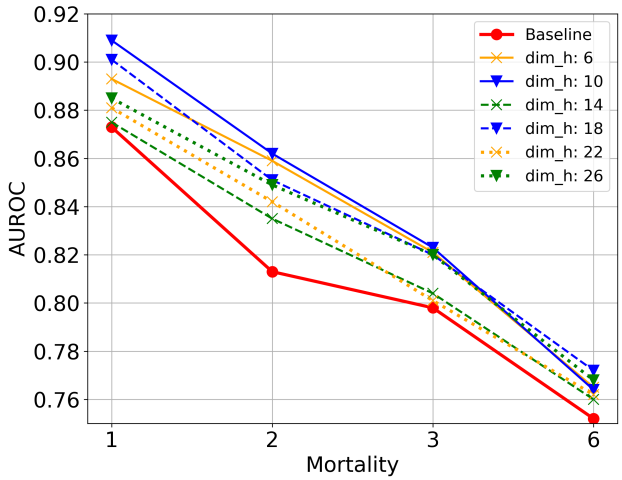


Fig. 5.8: Evaluation of TLAV changing the dimension of the codes.

5.5 Discussion

This chapter has explored a novel TL alternative based on sample and feature space augmentation based on TLCO and TLVA. Information was transferred from a massive data source and improved predictive mortality models in ESRD patients. The transferred information was extracted in the codes of both domains. It was shown that transferring knowledge from another data source directly improves the learning models using codes from AEs. The conducted experiments have shown that deep AEs extract better complex relationships for the available domains than mSDAs.

For the experiments related sample augmentation, it was found that TLCO provided an improvement, from 2-5% in AUROC, when both classes are transferred from AKI. It was evidenced that increasing just the imbalance in most models does not deteriorate the predictions' performance. Reducing data imbalance provides a considerable improvement for the learning models,

Tab. 5.4: Final comparison of the proposed TL framework. TLAV-TLCO is the cascade version of the TL proposed methods.

Mortality	Baseline	TLCO	TLAV	TLAV-HHTLM
1	0.873	0.891	0.909	**0.939**
2	0.813	0.845	0.862	**0.909**
3	0.798	0.838	0.823	**0.853**
6	0.752	**0.778**	0.765	0.764

although the predictive ability in the data increases considerably when both classes are included in the upsampling.

For the case of feature space augmentation, it was evidenced that increasing the information in ESRD with the AVG_{codes} improves the performance of the learning models even when other alternatives such as mSDA or HHTL are used. Moreover, TLAV was shown to generalize better than TLCO in predictive models for a 2-month mortality horizon. It was evidenced that the dominant parameter that controlled the performance of the learning models was the dimension of the codes. In the case of threshold ϵ, from the inclusion of 40% of the codes from AKI, it is enough to guarantee an increase in performance among 2-6% compared to the baseline models.

Finally, the results obtained showed that the proposed framework can improve the predictive capacity of mortality models in ESRD and that they can be complementary to each other. If these two are combined, the performance of these models increases considerably (6-11%). Such improvements in the performance of the mortality predictors could imply incorporating this type of solution into clinical setting brings us closer to incorporating data-driven solutions to support medical staff in the early detection of events such as mortality.

5.6 Conclusions

In this chapter, a TL learning approach for knowledge transfer between heterogeneous domains in a clinical setting has been proposed. This Framework has been designed to improve predictive models of mortality in ESRD patients, based on knowledge transfer of AKI patients from the massive MIMIC-III database. The proposed mechanisms were based on the manipulation of AEs codes. Samples were transferred and the feature space for ESRD data was increased. Conducted experiments have shown that the proposed framework performs better than other approaches in the literature. Using deep AEs codes for knowledge transfer is a considerable improvement in learning models. The proposed framework was shown to combat data imbalance on its way to improving predictive models. The combination of upsampling and feature space was shown to significantly improve the models by applying

these two solutions individually. The implementation of TL-based solutions in a clinical setting brings us closer to incorporating data-driven solutions to support medical staff in the early detection of events such as mortality.

6

Conclusions and future work

In this PhD dissertation we have explored and proposed several mechanisms for extracting knowledge from clinical data based on DL techniques. The main focus of this dissertation was related to the improvement of learning models in a known clinical environment, tackling inherent data issues such as missing information, imbalanced data, and low data volume.

First, we considered the improvement of mortality predictors for ESRD patients in a nephrology unit from a tertiary hospital. For such cohort of patients, we were able to improve the mortality prediction and we evaluated how different groups of variables that influenced the performance of the learning models. We then focused on addressing two main problems inherent to a clinical environment. In the first one, we addressed the challenge of MV imputation using a hybrid MI with a DL approach. As a second challenge, we focused on a TL environment to transfer knowledge from a level 1 health unit to the health unit collecting ESRD data and thus tackling various drawbacks in that cohort. In the following, we address the conclusions of this dissertation and some open issues to address as future work.

6.1 Conclusions

After motivating this thesis, in Chapter 2 we covered the necessary components to address the research objectives of this dissertation.

Chapter 3 was dedicated to mortality models in ESRD patients. This chapter integrated various data sources and evaluated their impact on learning models at different mortality windows. In addition, we evaluated several groups of variables to compare whether the groups chosen by the expert

staff were adequate for the learning models or whether it was possible to extract more knowledge through an ML approach such as RFE. As a result, we found that this approach distinguished the most relevant variables for the learning models and its performance was close to the model that used the entire feature space. This could allow expert staff to consider variables that were not previously taken into account for EBM-based models and open up new possibilities for risk factors research in this population. For temporal dependencies, we also found that although the performance of the models was better than the state-of-the-art, increasing the prediction window made the imbalance of samples more critical. Thus, the performance of the models decreased. In conclusion, we found that there is space for improvement in data-driven learning models, and such enhancements may be the key to changing the paradigm from EBM to data-based medicine.

Chapter 4 addresses one of the most common challenges in an ML environment, the lack of registers in data examples. In this chapter, we provided a hybrid approach that combined MI with the latent spaces of AEs to impute MVs. The presented approach showed that this novel integration improved the reconstruction of missing information at several MV ratios. Reconstruction using the AVG code was also robust to different MV generation mechanisms, such as MCARs, MARs and MNARs. In addition, we validated the proposed method with literature benchmarks. We found that it offers competent performances and is better than those methods with which it was compared in many scenarios. Regarding CKD, considering that this dataset had many more samples than the other clinical datasets, the proposed method performed better than its competitors in 16/17 cases. Thus, we conclude that the AVG code is a validated alternative that offers a significant reconstruction capacity to methods in the literature.

Chapter 5 is an extension of the work carried out in Chapter 3. This chapter addresses the issues encountered in the mortality prediction task for patients in ESRD. The challenges derived from this application were related to the volume of information and the imbalance of the classes. TL supported by DL tackled both challenges. We proposed TLCO for sample augmentation and TLAV for increasing the feature space in the target domain. We support the task in such domain with data related to AKI from MIMIC-III. We used the experience gained in Chapter 3 to define 4-month time windows and LSTM

parameters that best fit the data. AEs were used as information bridges for the transfer of knowledge, where the idea of the AVG code was taken up for TLAV. Both approaches offered gain in mortality prediction that surpassed the models in Chapter 3. This improvement was optimized once the proposed mechanisms were cascaded, going from 0.87 to a performance of 0.94 in AUROC. Finally, the proposals offered in terms of knowledge transfer between heterogeneous clinical data sources can serve as a basis for supporting clinical hypotheses based on multiple health entities.

6.2 Future work

The work performed in this Ph.D. dissertation can be extrapolated to pathologies structurally similar to those addressed in Chapter 3 and 4:

The future work of this dissertation can be divided into two components: experiments that remain to be validated and the long-term future of the findings of this dissertation. The pending experiments are listed below:

- Validate the predictive models of mortality in a prospective cohort of patients belonging to the renal unit of the ESRD data;
- Evaluate the incidence of the application of AVG code to the variables that were excluded, due to the number of VMs, in the study of mortality in ESRD patients;
- Consolidate and apply a pipeline that contemplates all the mechanisms offered in this dissertation to improve learning models in ESRD patients.

The long-term future contributions of this dissertation are related to the integration of the proposed approaches to other pathologies and the large-scale application of the proposed mechanisms. In particular, we will pursue:

- To use the knowledge generated in the study of kidney-related pathologies on a larger scale of patients, i.e., access regional databases to study similar pathologies on a larger scale;

- To integrate LSTM networks in the latent spaces of the AEs in order to explore temporal relationships in the data and evaluate their effectiveness in terms of information reconstruction with data with this type of dependencies, as in the case of mortality in ESRD;
- To incorporate regional data from Catalonia with the dual purpose of preparing both data to transfer knowledge between different sources and transfer mechanisms that support clinical hypotheses and propose generalized learning models supported by massive data sources

Bibliography

[AK19] Noura S Abul-Husn and Eimear E Kenny. “Personalized medicine and the power of electronic health records”. In: *Cell* 177.1 (2019), pp. 58–69 (cit. on pp. 10, 11).

[APP18] Giuseppe Aceto, Valerio Persico, and Antonio Pescapé. “The role of Information and Communication Technologies in healthcare: taxonomies, perspectives, and challenges”. In: *Journal of Network and Computer Applications* 107 (2018), pp. 125–154 (cit. on p. 1).

[Akb+19] Oguz Akbilgic, Yoshitsugu Obi, Praveen K. Potukuchi, et al. “Machine Learning to Identify Dialysis Patients at High Death Risk”. In: *Kidney International Reports* 4.9 (2019), pp. 1219–1229 (cit. on pp. 42, 56).

[AAQ17] Hamdan O. Alanazi, Abdul Hanan Abdullah, and Kashif Naseer Qureshi. “A Critical Review for Developing Accurate and Dynamic Predictive Models Using Machine Learning Methods in Medicine and Health Care”. In: *Journal of Medical Systems* (2017) (cit. on p. 42).

[Alm+19] Njoud Abdullah Almansour, Hajra Fahim Syed, Nuha Radwan Khayat, et al. “Neural network and support vector machine for the prediction of chronic kidney disease: A comparative study”. In: *Computers in Biology and Medicine* 109 (2019), pp. 101–111 (cit. on p. 42).

[AKK19] Hamed Alqahtani, Manolya Kavakli-Thorne, and Gulshan Kumar. “Applications of generative adversarial networks (gans): An updated review”. In: *Archives of Computational Methods in Engineering* (2019), pp. 1–28 (cit. on p. 61).

[BM17] Brett K. Beaulieu-Jones and Jason H. Moore. “Missing data imputation in the electronic health record using deeply learned autoencoders”. In: *Pacific Symposium on Biocomputing*. 2017 (cit. on p. 60).

[Bed+00] Srinivasan Beddhu, Frank J. Bruns, Melissa Saul, Patricia Seddon, and Mark L. Zeidel. “A simple comorbidity scale predicts clinical outcomes and costs in dialysis patients”. In: *American Journal of Medicine* (2000) (cit. on p. 42).

[BIF18] Houda Benhar, Ali Idri, and JL Fernández-Alemán. “Data preprocessing for decision making in medical informatics: potential and analysis”. In: *World conference on information systems and technologies*. Springer. 2018, pp. 1208–1218 (cit. on p. 12).

[Blu09] David Blumenthal. “Stimulating the adoption of health information technology”. In: *West Virginia Medical Journal* 105.3 (2009), pp. 28–30 (cit. on p. 10).

[BL01] Leo Breiman and Leo. “Random Forests”. In: *Machine Learning* 45.1 (2001), pp. 5–32 (cit. on p. 48).

[Bur+18] Alexandru Burlacu, Simonetta Genovesi, David Goldsmith, et al. “Bleeding in advanced CKD patients on antithrombotic medication–a critical appraisal”. In: *Pharmacological research* 129 (2018), pp. 535–543 (cit. on p. 1).

[Bur+19] Alexandru Burlacu, Simonetta Genovesi, Alberto Ortiz, et al. “Pros and cons of antithrombotic therapy in end-stage kidney disease: a 2019 update”. In: *Nephrology Dialysis Transplantation* 34.6 (2019), pp. 923–933 (cit. on p. 1).

[Byr+20] Michal Byra, Mei Wu, Xiaodong Zhang, et al. “Knee menisci segmentation and relaxometry of 3D ultrashort echo time cones MR imaging using attention U-Net with transfer learning”. In: *Magnetic resonance in medicine* 83.3 (2020), pp. 1109–1122 (cit. on p. 76).

[CR89] William M. Campion and Donald B. Rubin. “Multiple Imputation for Nonresponse in Surveys”. In: *Journal of Marketing Research* (1989) (cit. on p. 60).

[Çel+14] Güner Çelik, Ömer K Baykan, Yakup Kara, and Hülya Tireli. “Predicting 10-day mortality in patients with strokes using neural networks and multivariate statistical methods”. In: *Journal of Stroke and Cerebrovascular Diseases* 23.6 (2014), pp. 1506–1512 (cit. on p. 42).

[Cha+19] Angela Y. Chang, Krycia Cowling, Angela E. Micah, et al. “Past, present, and future of global health financing: a review of development assistance, government, out-of-pocket, and other private spending on health for 195 countries, 1995–2050”. In: *The Lancet* 393.10187 (2019), pp. 2233–2260 (cit. on p. 1).

[Che+12] Minmin Chen, Zhixiang Xu, Kilian Weinberger, and Fei Sha. “Marginalized denoising autoencoders for domain adaptation”. In: *arXiv preprint arXiv:1206.4683* (2012) (cit. on p. 78).

[Chi+18] Travers Ching, Daniel S Himmelstein, Brett K Beaulieu-Jones, et al. “Opportunities and obstacles for deep learning in biology and medicine”. In: *Journal of The Royal Society Interface* 15.141 (2018), p. 20170387 (cit. on pp. 10, 60, 76).

[Cho+21] Tejalal Choudhary, Vipul Mishra, Anurag Goswami, and Jagannathan Sarangapani. “A transfer learning with structured filter pruning approach for improved breast cancer classification on point-of-care devices”. In: *Computers in Biology and Medicine* 134 (2021), p. 104432 (cit. on p. 76).

[Cis+13] Federico Cismondi, André S Fialho, Susana M Vieira, et al. “Missing data in medical databases: Impute, delete or classify?” In: *Artificial Intelligence in Medicine* 58.1 (2013), pp. 63–72 (cit. on p. 60).

[Coc97] Alistair J Cochran. “Prediction of outcome for patients with cutaneous melanoma”. In: *Pigment cell research* 10.3 (1997), pp. 162–167 (cit. on p. 9).

[Coo+97] Gregory F Cooper, Constantin F Aliferis, Richard Ambrosino, et al. “An evaluation of machine-learning methods for predicting pneumonia mortality”. In: *Artificial intelligence in medicine* 9.2 (1997), pp. 107–138 (cit. on p. 9).

[CV95] Corinna Cortes and Vladimir Vapnik. “Support-vector networks”. In: *Machine learning* 20.3 (1995), pp. 273–297 (cit. on pp. 26, 28).

[Cou+09] Cécile Couchoud, Michel Labeeuw, Olivier Moranne, et al. “A clinical score to predict 6-month prognosis in elderly patients starting dialysis for end-stage renal disease”. In: *Nephrology Dialysis Transplantation* (2009) (cit. on p. 42).

[Cov+18] Adrian Covic, Simonetta Genovesi, Patrick Rossignol, et al. “Practical issues in clinical scenarios involving CKD patients requiring antithrombotic therapy in light of the 2017 ESC guideline recommendations”. In: *BMC medicine* 16.1 (2018), pp. 1–11 (cit. on p. 1).

[DAg+08a] Ralph B. D’Agostino, Ramachandran S. Vasan, Michael J. Pencina, et al. “General cardiovascular risk profile for use in primary care: The Framingham heart study”. In: *Circulation* (2008) (cit. on p. 42).

[DAg+08b] Ralph B D’Agostino Sr, Ramachandran S Vasan, Michael J Pencina, et al. “General cardiovascular risk profile for use in primary care: the Framingham Heart Study”. In: *Circulation* 117.6 (2008), pp. 743–753 (cit. on p. 12).

[Dar+20] Shaveta Dargan, Munish Kumar, Maruthi Rohit Ayyagari, and Gulshan Kumar. “A survey of deep learning and its applications: a new paradigm to machine learning”. In: *Archives of Computational Methods in Engineering* 27.4 (2020), pp. 1071–1092 (cit. on p. 26).

[DME18] Burcu F Darst, Kristen C Malecki, and Corinne D Engelman. “Using recursive feature elimination in random forest to account for correlated variables in high dimensional data”. In: *BMC genetics* 19.1 (2018), p. 65 (cit. on p. 49).

[Des+17] Thomas Desautels, Jacob Calvert, Jana Hoffman, et al. “Using transfer learning for improved mortality prediction in a data-scarce hospital setting”. In: *Biomedical informatics insights* 9 (2017), p. 1178222617712994 (cit. on p. 76).

[Dil+19] Gerhard-Paul Diller, Aleksander Kempny, Sonya V Babu-Narayan, et al. “Machine learning algorithms estimating prognosis and guiding therapy in adult congenital heart disease: data from a single tertiary centre including 10 019 patients”. In: *European heart journal* 40.13 (2019), pp. 1069–1077 (cit. on p. 20).

[Doi+15] Toshiki Doi, Suguru Yamamoto, Takatoshi Morinaga, et al. “Risk score to predict 1-year mortality after haemodialysis initiation in patients with stage 5 chronic kidney disease under predialysis nephrology care”. In: *PloS one* 10.6 (2015) (cit. on p. 42).

[DO02] Stephan Dreiseitl and Lucila Ohno-Machado. “Logistic regression and artificial neural network classification models: A methodology review”. In: *Journal of Biomedical Informatics* (2002) (cit. on p. 42).

[DG17] Dheeru Dua and Casey Graff. *UCI Machine Learning Repository*. 2017 (cit. on p. 65).

[Duh+03] Alain Duhamel, MC Nuttens, Patrick Devos, Monique Picavet, and Régis Beuscart. “A preprocessing method for improving data mining techniques. Application to a large medical diabetes database”. In: *The new navigators: from professionals to patients*. IOS Press, 2003, pp. 269–274 (cit. on p. 13).

[EVM20] Hossein Estiri, Sebastien Vasey, and Shawn N Murphy. “Generative transfer learning for measuring plausibility of EHR diagnosis records”. In: *Journal of the American Medical Informatics Association* 28.3 (Oct. 2020), pp. 559–568. eprint: `https://academic.oup.com/jamia/article-pdf/28/3/559/36428626/ocaa215.pdf` (cit. on p. 76).

[EGF11] Konstantinos P Exarchos, Yorgos Goletsis, and Dimitrios I Fotiadis. "Multiparametric decision support system for the prediction of oral cancer reoccurrence". In: *IEEE Transactions on Information Technology in Biomedicine* 16.6 (2011), pp. 1127–1134 (cit. on p. 9).

[FP+17] Meherwar Fatima, Maruf Pasha, et al. "Survey of machine learning algorithms for disease diagnostic". In: *Journal of Intelligent Learning Systems and Applications* 9.01 (2017), p. 1 (cit. on p. 26).

[FPS96] Usama Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth. "From data mining to knowledge discovery in databases". In: *AI magazine* 17.3 (1996), pp. 37–37 (cit. on p. 11).

[Fer+19] Patrizia Ferroni, Fabio M Zanzotto, Silvia Riondino, et al. "Breast cancer prognosis using a machine learning approach". In: *Cancers* 11.3 (2019), p. 328 (cit. on p. 20).

[Frö+18] Holger Fröhlich, Rudi Balling, Niko Beerenwinkel, et al. "From hype to reality: data science enabling personalized medicine". In: *BMC medicine* 16.1 (2018), pp. 1–15 (cit. on p. 20).

[GLD00] Dragan Gamberger, Nada Lavrac, and Saso Dzeroski. "Noise detection and elimination in data preprocessing: experiments in medical domains". In: *Applied artificial intelligence* 14.2 (2000), pp. 205–223 (cit. on p. 12).

[Gia+18] Milena A. Gianfrancesco, Suzanne Tamang, Jinoos Yazdany, and Gabriela Schmajuk. *Potential Biases in Machine Learning Algorithms Using Electronic Health Record Data*. 2018 (cit. on p. 60).

[Gof+14] C Goff David, M Lloyd-Jones Donald, G Bennett, et al. "ACC/AHA guideline on the assessment of cardiovascular risk". In: *Circulation* 129.25_suppl_2 (2014), S49–S73 (cit. on pp. 12, 42).

[GW18] Lovedeep Gondara and Ke Wang. "MIDA: Multiple imputation using denoising autoencoders". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. 2018. arXiv: 1705.02737 (cit. on pp. 60, 61, 63).

[GBC16] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016 (cit. on p. 10).

[Goo+14] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, et al. "Generative adversarial nets". In: *Advances in neural information processing systems* 27 (2014) (cit. on pp. 31, 60).

[GT12] Geoff Gordon and Ryan Tibshirani. “Karush-kuhn-tucker conditions”. In: *Optimization* 10.725/36 (2012), p. 725 (cit. on p. 28).

[Hai+17] Guo Haixiang, Li Yijing, Jennifer Shang, et al. “Learning from class-imbalanced data: Review of methods and applications”. In: *Expert Systems with Applications* 73 (2017), pp. 220–239 (cit. on p. 77).

[Har+19] Hrayr Harutyunyan, Hrant Khachatrian, David C Kale, Greg Ver Steeg, and Aram Galstyan. “Multitask learning and benchmarking with clinical time series data”. In: *Scientific data* 6.1 (2019), pp. 1–18 (cit. on p. 20).

[He+19] Jianxing He, Sally L Baxter, Jie Xu, et al. “The practical implementation of artificial intelligence technologies in medicine”. In: *Nature medicine* 25.1 (2019), pp. 30–36 (cit. on p. 20).

[Hin+12] Geoffrey Hinton, Li Deng, Dong Yu, et al. “Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups”. In: *IEEE Signal processing magazine* 29.6 (2012), pp. 82–97 (cit. on p. 2).

[Hip+08] Julia Hippisley-Cox, Carol Coupland, Yana Vinogradova, et al. “Predicting cardiovascular risk in England and Wales: prospective derivation and validation of QRISK2”. In: *Bmj* 336.7659 (2008), pp. 1475–1482 (cit. on pp. 12, 42).

[HJ14] Andreas Holzinger and Igor Jurisica. “Knowledge discovery and data mining in biomedical informatics: The future is in integrative, interactive machine learning solutions”. In: *Interactive knowledge discovery and data mining in biomedical informatics*. Springer, 2014, pp. 1–18 (cit. on p. 20).

[Hsi+11] Eileen Hsich, Eiran Z Gorodeski, Eugene H Blackstone, Hemant Ishwaran, and Michael S Lauer. “Identifying important risk factors for survival in patient with systolic heart failure using random survival forests”. In: *Circulation: Cardiovascular Quality and Outcomes* 4.1 (2011), pp. 39–45 (cit. on p. 9).

[Ink+14] Lesley A. Inker, Brad C. Astor, Chester H. Fox, et al. “KDOQI US commentary on the 2012 KDIGO clinical practice guideline for the evaluation and management of CKD”. In: *American Journal of Kidney Diseases* (2014) (cit. on pp. 66, 87).

[Ion+13] Jasmine Ion Titapiccolo, Manuela Ferrario, Sergio Cerutti, et al. "Artificial intelligence models to stratify cardiovascular risk in incident hemodialysis patients". In: *Expert Systems with Applications* (2013) (cit. on p. 42).

[Ish+08] Hemant Ishwaran, Udaya B Kogalur, Eugene H Blackstone, and Michael S Lauer. "Random survival forests". In: *The annals of applied statistics* 2.3 (2008), pp. 841–860 (cit. on p. 9).

[Iss+14] Keiji Isshiki, Toshiki Nishio, Motohide Isono, et al. "Glycated albumin predicts the risk of mortality in type 2 diabetic patients on hemodialysis: evaluation of a target level for improving survival". In: *Therapeutic Apheresis and Dialysis* 18.5 (2014), pp. 434–442 (cit. on p. 56).

[Jia+17] Fei Jiang, Yong Jiang, Hui Zhi, et al. "Artificial intelligence in healthcare: past, present and future". In: *Stroke and vascular neurology* 2.4 (2017) (cit. on p. 26).

[Joh+16a] Alistair E.W. Johnson, Tom J. Pollard, Lu Shen, et al. "MIMIC-III, a freely accessible critical care database". In: *Scientific Data* (2016) (cit. on p. 65).

[Joh+16b] Alistair EW Johnson, Tom J Pollard, Lu Shen, et al. "MIMIC-III, a freely accessible critical care database". In: *Scientific data* 3 (2016), p. 160035 (cit. on p. 87).

[KB14] Diederik P Kingma and Jimmy Ba. "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980* (2014) (cit. on p. 35).

[KB15] Diederik P. Kingma and Jimmy Lei Ba. "Adam: A method for stochastic optimization". In: *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*. 2015. arXiv: 1412.6980 (cit. on p. 67).

[Kom+18] Matthieu Komorowski, Leo A Celi, Omar Badawi, Anthony C Gordon, and A Aldo Faisal. "The artificial intelligence clinician learns optimal treatment strategies for sepsis in intensive care". In: *Nature medicine* 24.11 (2018), pp. 1716–1720 (cit. on p. 20).

[Kon01] Igor Kononenko. "Machine learning for medical diagnosis: history, state of the art and perspective". In: *Artificial Intelligence in medicine* 23.1 (2001), pp. 89–109 (cit. on p. 9).

[KH92] Anders Krogh and John A Hertz. “A simple weight decay can improve generalization”. In: *Advances in neural information processing systems*. 1992, pp. 950–957 (cit. on p. 35).

[LB+95] Yann LeCun, Yoshua Bengio, et al. “Convolutional networks for images, speech, and time series”. In: *The handbook of brain theory and neural networks* 3361.10 (1995), p. 1995 (cit. on p. 31).

[Liu+10] Jiannong Liu, Zhi Huang, David T. Gilbertson, Robert N. Foley, and Allan J. Collins. “An improved comorbidity index for outcome analyses among dialysis patients”. In: *Kidney International* (2010) (cit. on p. 42).

[Lop+21] Ricardo R. Lopes, Hidde Bleijendaal, Lucas A. Ramos, et al. “Improving electrocardiogram-based detection of rare genetic heart disease using transfer learning: An application to phospholamban p.Arg14del mutation carriers”. In: *Computers in Biology and Medicine* 131 (2021), p. 104262 (cit. on p. 76).

[LTS18] Valerie A. Luyckx, Marcello Tonelli, and John W. Stanifer. “The global burden of kidney disease and the sustainable development goals”. In: *Bulletin of the World Health Organization* (2018) (cit. on p. 42).

[Lv+21] Jun Lv, Guangyuan Li, Xiangrong Tong, et al. “Transfer learning enhanced generative adversarial networks for multi-channel MRI reconstruction”. In: *Computers in Biology and Medicine* 134 (2021), p. 104504 (cit. on p. 76).

[Mac+18] E Macias, A Morell, J Serrano, and JL Vicario. “Knowledge extraction based on wavelets and DNN for classification of physiological signals: Arousals case”. In: *2018 Computing in Cardiology Conference (CinC)*. Vol. 45. IEEE. 2018, pp. 1–4 (cit. on p. 77).

[Mac+20] Edwar Macias, Antoni Morell, Javier Serrano, Jose Lopez Vicario, and Jose Ibeas. “Mortality prediction enhancement in end-stage renal disease: A machine learning approach”. In: *Informatics in Medicine Unlocked* 19 (2020), p. 100351 (cit. on pp. 18, 87, 89).

[Mac+19] Edwar Macias Toro, Guillem Boquet, Javier Serrano, et al. “Novel Imputing Method and Deep Learning Techniques for Early Prediction of Sepsis in Intensive Care Units”. In: *2019 Computing in Cardiology Conference (CinC)*. 2019 (cit. on pp. 60, 77).

[Mad+13] Dharanija Madhavan, Katarina Cuk, Barbara Burwinkel, and Rongxi Yang. “Cancer diagnosis and prognosis decoded by blood-based circulating microRNA signatures”. In: *Frontiers in genetics* 4 (2013), p. 116 (cit. on p. 9).

[Maq+19] Muazzam Maqsood, Faria Nazir, Umair Khan, et al. “Transfer Learning Assisted Classification and Detection of Alzheimer’s Disease Stages Using 3D MRI Scans”. In: *Sensors* 19.11 (2019) (cit. on p. 76).

[Mar+10] Daniel S Marcus, Anthony F Fotenos, John G Csernansky, John C Morris, and Randy L Buckner. “Open access series of imaging studies: longitudinal MRI data in nondemented and demented older adults”. In: *Journal of cognitive neuroscience* 22.12 (2010), pp. 2677–2684 (cit. on p. 76).

[Mar+14] Alberto Martínez-Castelao, José L. Górriz, Jordi Bover, et al. “Documento de consenso para la detección y manejo de la enfermedad renal crónica”. In: *Semergen* 40.8 (2014), pp. 441–459 (cit. on p. 42).

[Mau+08] Joan M. Mauri, Montse Clèries, Emili Vela, and Catalan Renal Registry. “Design and validation of a model to predict early mortality in haemodialysis patients”. In: *Nephrology Dialysis Transplantation* (2008) (cit. on pp. 42, 56).

[MP43] Warren S McCulloch and Walter Pitts. “A logical calculus of the ideas immanent in nervous activity”. In: *The bulletin of mathematical biophysics* 5.4 (1943), pp. 115–133 (cit. on p. 10).

[MMC15] Elizabeth A McGlynn, Kathryn M McDonald, and Christine K Cassel. “Measurement is essential for improving diagnosis and reducing diagnostic error: a report from the Institute of Medicine”. In: *Jama* 314.23 (2015), pp. 2501–2502 (cit. on p. 1).

[Mil+19] Rebecca Miller, Dmitry Tumin, Jennifer Cooper, Don Hayes, and Joseph D. Tobias. “Prediction of mortality following pediatric heart transplant using machine learning algorithms”. In: *Pediatric Transplantation* (2019) (cit. on p. 42).

[MB19] Yoav Mintz and Ronit Brodie. “Introduction to artificial intelligence in medicine”. In: *Minimally Invasive Therapy & Allied Technologies* 28.2 (2019), pp. 73–81 (cit. on p. 9).

[Mir+16] E. M. Mirkes, T. J. Coats, J. Levesley, and A. N. Gorban. “Handling missing data in large healthcare dataset: A case study of unknown trauma outcomes”. In: *Computers in Biology and Medicine* (2016). arXiv: 1604.00627 (cit. on p. 60).

[MS19] Stefania Montani and Manuel Striani. “Artificial intelligence in clinical decision support: a focused literature survey”. In: *Yearbook of medical informatics* 28.1 (2019), p. 120 (cit. on p. 1).

[Moo+15] Karel GM Moons, Douglas G Altman, Johannes B Reitsma, et al. “Transparent Reporting of a multivariable prediction model for Individual Prognosis or Diagnosis (TRIPOD): explanation and elaboration”. In: *Annals of internal medicine* 162.1 (2015), W1–W73 (cit. on p. 12).

[Mot+17] Manish Motwani, Damini Dey, Daniel S Berman, et al. “Machine learning for prediction of all-cause mortality in patients with suspected coronary artery disease: a 5-year multicentre prospective registry analysis”. In: *European heart journal* 38.7 (2017), pp. 500–507 (cit. on p. 9).

[NKW18] Stefano Nembrini, Inke R König, and Marvin N Wright. “The revival of the Gini importance?” In: *Bioinformatics* 34.21 (2018), pp. 3711–3718 (cit. on p. 30).

[Ote+12] M Otero-López, Juan C Martınez-Ocaña, Loreley Betancourt-Castellanos, Eleonora Rodrıguez-Salazar, and Manuel Garcıa-Garcıa. “Two prognostic scores for early mortality and their clinical applicability in elderly patients on haemodialysis: poor predictive success in individual patients”. In: *Nefrologıa (English Edition)* 32.2 (2012), pp. 213–220 (cit. on p. 42).

[PY10] Sinno Jialin Pan and Qiang Yang. “A Survey on Transfer Learning”. In: *IEEE Transactions on Knowledge and Data Engineering* 22.10 (2010), pp. 1345–1359 (cit. on p. 76).

[Pér+15] Joaquın Pérez, Emmanuel Iturbide, Vıctor Olivares, et al. “A data preparation methodology in data mining applied to mortality population databases”. In: *New contributions in information systems and technologies*. Springer, 2015, pp. 1173–1182 (cit. on p. 13).

[Pic+21] Francesco Piccialli, Vittorio Di Somma, Fabio Giampaolo, Salvatore Cuomo, and Giancarlo Fortino. “A survey on deep learning in medicine: Why, how and when?” In: *Information Fusion* 66 (2021), pp. 111–137 (cit. on p. 76).

[Pla98] John Platt. “Sequential minimal optimization: A fast algorithm for training support vector machines”. In: (1998) (cit. on p. 28).

[Pre12] Lutz Prechelt. “Early stopping - But when?” In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (2012) (cit. on p. 67).

[Qui14] J Ross Quinlan. *C4. 5: programs for machine learning*. Elsevier, 2014 (cit. on p. 29).

[RK15] Connie M. Rhee and Csaba P. Kovesdy. “Spotlight on CKD deaths—increasing mortality worldwide”. In: *Nature Reviews Nephrology* (2015) (cit. on p. 42).

[Rid+07] Paul M Ridker, Julie E Buring, Nader Rifai, and Nancy R Cook. “Development and validation of improved algorithms for the assessment of global cardiovascular risk in women: the Reynolds Risk Score”. In: *Jama* 297.6 (2007), pp. 611–619 (cit. on pp. 12, 42).

[Ros+16] Elsie Gyang Ross, Nigam H Shah, Ronald L Dalman, et al. “The use of machine learning for the identification of peripheral artery disease and future mortality risk”. In: *Journal of vascular surgery* 64.5 (2016), pp. 1515–1522 (cit. on p. 42).

[Rub76] Donald B. Rubin. “Inference and missing data”. In: *Biometrika* (1976) (cit. on p. 17).

[RHW85] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. *Learning internal representations by error propagation*. Tech. rep. California Univ San Diego La Jolla Inst for Cognitive Science, 1985 (cit. on p. 31).

[RHW86a] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. “Learning representations by back-propagating errors”. In: *nature* 323.6088 (1986), pp. 533–536 (cit. on p. 31).

[RHW86b] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. “Learning representations by back-propagating errors”. In: *nature* 323.6088 (1986), pp. 533–536 (cit. on p. 34).

[Rus+15] Olga Russakovsky, Jia Deng, Hao Su, et al. “Imagenet large scale visual recognition challenge”. In: *International journal of computer vision* 115.3 (2015), pp. 211–252 (cit. on p. 2).

[Sac97] David L Sackett. “Evidence-based medicine”. In: *Seminars in perinatology*. Vol. 21. 1. Elsevier. 1997, pp. 3–5 (cit. on p. 1).

[SS14] Barna Saha and Divesh Srivastava. “Data quality: The other face of Big Data”. In: *2014 IEEE 30th International Conference on Data Engineering*. 2014, pp. 1294–1297 (cit. on p. 12).

[SS16] Asif Salekin and John Stankovic. “Detection of chronic kidney disease and selecting important predictive attributes”. In: *2016 IEEE International Conference on Healthcare Informatics (ICHI)*. IEEE. 2016, pp. 262–270 (cit. on p. 42).

[San+19] Miriam Seoane Santos, Ricardo Cardoso Pereira, Adriana Fonseca Costa, et al. “Generating synthetic missing data: A review by missing mechanism”. In: *IEEE Access* (2019) (cit. on pp. 68, 69).

[Shi+21] Benjamin Shickel, Anis Davoudi, Tezcan Ozrazgat-Baslanti, et al. “Deep Multi-Modal Transfer Learning for Augmented Patient Acuity Assessment in the Intelligent ICU”. In: *Frontiers in Digital Health* 3 (2021), p. 11 (cit. on p. 76).

[Shu19] Jessica Germaine Shull. “Digital health and the state of interoperable electronic health records”. In: *JMIR medical informatics* 7.4 (2019), e12712 (cit. on p. 11).

[Sri+14] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. “Dropout: A simple way to prevent neural networks from overfitting”. In: *Journal of Machine Learning Research* (2014) (cit. on p. 67).

[Ste+09] Jonathan A.C. Sterne, Ian R. White, John B. Carlin, et al. *Multiple imputation for missing data in epidemiological and clinical research: Potential and pitfalls*. 2009 (cit. on p. 60).

[Sun+07] Yijun Sun, Steve Goodison, Jian Li, Li Liu, and William Farmerie. “Improved breast cancer prognosis through the combination of clinical and genetic markers”. In: *Bioinformatics* 23.1 (2007), pp. 30–37 (cit. on p. 9).

[SVL14] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. “Sequence to sequence learning with neural networks”. In: *Advances in neural information processing systems*. 2014, pp. 3104–3112 (cit. on p. 2).

[Vin+10] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, et al. “Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion.” In: *Journal of machine learning research* 11.12 (2010) (cit. on p. 78).

[Wag+11] Martin Wagner, David Ansell, David M. Kent, et al. “Predicting mortality in incident dialysis patients: An analysis of the United Kingdom renal registry”. In: *American Journal of Kidney Diseases* (2011) (cit. on p. 56).

[Wan+15] Guanjin Wang, Kin Man Lam, Zhaohong Deng, and Kup Sze Choi. "Prediction of mortality after radical cystectomy for bladder cancer by machine learning techniques". In: *Computers in Biology and Medicine* (2015) (cit. on p. 42).

[WLR19] Ping Wang, Yan Li, and Chandan K Reddy. "Machine learning for survival analysis: A survey". In: *ACM Computing Surveys (CSUR)* 51.6 (2019), pp. 1–36 (cit. on p. 26).

[Wer94] Paul John Werbos. *The roots of backpropagation: from ordered derivatives to neural networks and political forecasting*. Vol. 1. John Wiley & Sons, 1994 (cit. on p. 34).

[YJV18] Jinsung Yoon, James Jordon, and Mihaela Van Der Schaar. "GAIN: Missing data imputation using generative adversarial nets". In: *35th International Conference on Machine Learning, ICML 2018*. 2018. arXiv: 1806.02920 (cit. on pp. 60, 66).

[Yu+19] Ying Yu, Min Li, Liangliang Liu, Yaohang Li, and Jianxin Wang. "Clinical big data and deep learning: Applications, challenges, and future outlooks". In: *Big Data Mining and Analytics* 2.4 (2019), pp. 288–305 (cit. on p. 76).

[Yue+20] Lin Yue, Dongyuan Tian, Weitong Chen, Xuming Han, and Minghao Yin. "Deep learning for heterogeneous medical data analysis". In: *World Wide Web* 23.5 (2020), pp. 2715–2737 (cit. on p. 10).

[ZPT19] Joey Tianyi Zhou, Sinno Jialin Pan, and Ivor W. Tsang. "A deep learning framework for Hybrid Heterogeneous Transfer Learning". In: *Artificial Intelligence* 275 (2019), pp. 310–328 (cit. on pp. 80, 88).

Webpages

[@Com19] European Comission. *eHealth adoption in primary healthcare in the EU is on the rise*. 2019. URL: https://digital-strategy.ec.europa.eu/en/library/ehealth-adoption-primary-healthcare-eu-rise (visited on Sept. 2, 2021) (cit. on p. 11).