Transparent approach based on deep learning and multiagent argumentation for hypertension management

Naziha Sendi

▶ **To cite this version:**

Naziha Sendi. Transparent approach based on deep learning and multiagent argumentation for hypertension management. Artificial Intelligence [cs.AI]. Université Paris-Saclay, 2020. English. NNT : 2020UPASG036 . tel-03311681

HAL Id: tel-03311681
https://theses.hal.science/tel-03311681

Submitted on 2 Aug 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

université
PARIS-SACLAY

Transparent approach based on deep learning and multiagent argumentation for hypertension management

Thèse de doctorat de l'Université Paris-Saclay

École doctorale n° 580 Sciences et technologies de l'information et de la communication (STIC)
Spécialité de doctorat: Informatique
Unité de recherche: IBISC, 91020, Evry-Courcouronnes, France
Référent: Université d'Evry-Val-d'Essonne

Thèse présentée et soutenue à Evry, le 16 décembre 2020, par

Naziha SENDI

Composition du jury:

Sylvie Després Professeur, université Paris 13	Présidente
Joao Marques-Silva Professeur, université de Toulouse	Rapporteur
Souhila KACI Professeur, université de Montpelier	Rapportrice
Mohamed Elati Professeur, université de Lille	Examinateur
Serenella Cerrito Professeur, université Paris-Saclay, univ. Evry	Examinatrice
Etienne Colle Professeur, Université Paris-Saclay, univ. Evry	Directeur
Farida Zehraoui Maître de Conférences, université Paris-Saclay, univ. Evry	Coencadrante
Nadia Abchiche-Mimouni Maître de Conférences, Université Paris-Saclay, univ. Evry	Coencadrante
François Teboul Médecin urgentist	Invité
Jean-Pierre Albinet C.T.O, Visiomed Group	Invité
Pinar Yolum Professeur, Université Utrecht	Invité

Thèse de doctorat

NNT: 2020UPASG036

Acknowledgements

First and foremost, praises and thanks to the God, the Almighty, for His showers of blessings throughout my research work to complete the research successfully.

After writing this research, I am convinced that the thesis is far from being a solitary task. Indeed, I would never have been able to carry out this doctoral work without the support of many people whose generosity, good humor and interest in my research allowed me to progress in this field.

I would like to express my deep and sincere gratitude to my research supervisor, Etienne Colle, professor at the University of Evry, for the trust he gave me by agreeing to supervise this thesis, for his many advices and for all the hours he devoted to directing this research project.

I would like to thank my thesis supervisors, Farida Zehraoui and Nadia Abchiche-Mimouni, for giving me the opportunity to work on this subject. Thanks to their expertises and advices, I was able to understand and immerse myself in this ambitious project combining two of my areas of interest. In addition, they supported me when I encountered difficulties and were able to guide me to complete my work. I would like to tell them how much I appreciated their availability and unwavering respect for the tight deadlines for proofreading the documents I sent them. Finally, I was extremely sensitive to their human qualities of listening and understanding throughout this doctoral work.

I would like to express my gratitude to Dr. François Teboul, for his involvement in the project, on the issue of medical concepts. He taught me a lot, I appreciated his enthusiasm and sympathy.

I would like to thank the reporters of this thesis Joao Marques-Silva, professer at Université de Toulouse and Souhila KACI, professer at Université de Montpelier, for the interest they have shown to my work.
I am extending my thanks to my manager Jean-Pierre Albinet for his support during my research work.

I express my special thanks to Antoine Jouannais for his encouragement and support.

I also thank the entire team at the IBISC laboratory and Visiomed Group/BewellConnect for providing me the favorable conditions for the realization of my thesis.

I am extremely grateful to my parents for their love, prayers, caring and sacrifices for educating and preparing me for my future. I am very much thankful to my husband for his love, understanding, and continuing support to complete this research work. Also I express my thanks to my sisters and my brothers for their support.

Finally, my thanks go to all the people who have supported me to complete the research work directly or indirectly.

Foreword

This thesis was carried out with the support of the National Agency for Research and Technology (ANRT) as part of a CIFRE contract (n2016 / 1554). Visiomed Group hosted me for the realization of the work. Academic supervision was provided by the laboratory IBISC of the University Evry, University Paris Saclay.

Visiomed Group is one of the French leaders in new generation medical electronics. The laboratory develops and markets innovative health products in the fields of self-diagnosis for medical use and well-being. In 2014, Visiomed Group launched the Bewell Connect brand, this brand includes a range of smart connected objects associated with a unique interpretation and personalized monitoring platform. Visiomed Group was the first French company to enter the Internet of health objects market by launching a unique range of connected health objects under the Bewell Connect brand. Since then, BewellConnect offers innovative solutions across the telemedicine value chain that improve care, patient monitoring and care coordination.

Combining multidisciplinary, fundamental, and applied research, and anchored in Information Science and Technology, the IBISC (Computing, Integrative Biology and Complex Systems) laboratory, is positioned as a strong pole in Ile de France. The research activities carried out within the IBISC laboratory deal with the modeling, design, simulation, and validation of complex systems. The systems considered are both biological systems and artificial systems (robots, drones, intelligent vehicles). IBISC aims to develop methods, formalisms, and realizations for the understanding of complex systems.

The two organisms were particularly interested in the development of decision support systems for doctors and patients with Hypertension. In this context, we collaborate to set up an individualized system for hypertension management according to different profiles. This will allow patients to take care of themselves and to have access to information and advice adapted to their profile.

Contents

List of Figures

List of Tables

Acronyms

AB Alpha Blockers

AE Autoencoder

AI Artificial Intelligence

BB Beta Blockers

CA Calcium Antagonist

CDS Clinical Decision System

CNN Convolutional Neural Network

DBP Diastolic Blood Pressure

DI Diuritics

DL Deep Learning

DMLP Deep multilayer network

DNNs Deep Neural Networks

DT Decision Tree

EBM Evidence-based medicine

EHR Electronic Health Records

FC-FFNN Fully connected feed-forward neural networks

GRU Gated Recurrent Units

ICE ACE Inhibitors

kNN k-Nearest Neighbors

LIME Local Interpretable Model-Agnostic Explanations

LSTM Long Short-Term Memory

MAS Multiagent system

ML Machine Learning

MLPs Multilayer Perceptrons

MS-LSTM Multisources LSTM

PBM Proof-based medicine

RNN Recurrent Neural Networks

SAR Sartans

SBP Systolic Blood Pressure

SVM Support Vector Machines

Chapter 1

Introduction and Context

Contents

1.1 General Introduction

Every person is unique and, in many ways, so are diseases. Yet, it is important to consider the profile of each patient in order to provide a deeper interpretation of an individual's characteristics and understanding of disease mechanisms which will lead to better treatment and prevention. This is what we call personalized medicine, which refers to the tailoring of medical treatment to the individual characteristics of each patient to classify individuals into subpopulations that differ in their susceptibility to a particular disease or their response to a specific treatment. Preventative or therapeutic interventions can then be concentrated on those who will benefit, sparing expense and side effects for those who will not.
Diagnostic and therapeutic resources have recently made a considerable progress. With the advances made in Artificial Intelligence (AI) [193] and the massive health records, new deals are to switch from population medicine to individual therapies, from curing the disease to preventing it. The aims are to reduce the frequency of diseases and to detect them earlier and earlier. Thus, it has become possible to follow-up healthcare which gives rise to personalization in medicine. This new approach, called "individualized and predictive" medicine, has become a great field opened by new techniques. The hope is that reading and interpreting the characteristics of an individual allows to better understand the mechanisms of the diseases, to better treat and prevent them.
In the same context, when it comes to predicting a risk for a person this is called "predictive medicine". This should allow to evaluate the risk of developing disease for individuals and to manage the health of persons according to their profiles. The predicted risks of a specific diagnosis or health outcome can be used by patients and doctors to support decisions.

The identification of the patients profiles allows to prescribe more targeted and therefore more effective drugs which means "The right drug, to the right person, at the right time". Such a personalization does not mean that drugs are created for a single individual. Rather, it translates into the ability to classify individuals into sub-populations characterized by predisposition to certain diseases or by response to a particular treatment. This concerns both preventive and therapeutic measures.
Medicine tends to consider patient as a whole: a body and a mind being thinking and interacting with his environment. It is then a question of adapting a treatment according to the individual characteristics of each patient. Thus, we speak about "4P" medicine for [33]:

- Personalized [240]: everyone is unique, we are interested in the personal profile of the individual (genetic, environmental, etc.);
- Preventive: through health education, we aim to reduce the risk of disease (primary prevention), promote early detection (secondary prevention) and improve the quality of life of sick people (tertiary prevention). "Well-being" is at the center of these different processes;
- Predictive [132]: By establishing a personalized mapping of risk factors and protective elements of a person's health, we can assess the risk of developing a disease and offer the most appropriate treatments;
- Participatory [132]: patients are the actors of their health and their care. They are now considered "expert patients", with theoretical knowledge and subjective knowledge from their own experience.

In real life, doctors have always felt that their decisions were based on evidence and proof. But, with 4P medicine, it is difficult for clinicians to keep up to date to new information relevant to their practice and to integrate their clinical experience. Therefore, we address Evidence-based medicine (EBM) [228] or Proof-based medicine (PBM) [295]. After 4P medicine (personalized, preventive, predictive and participatory), 5P medicine (addition of Proof medicine) is based on the use of the best evidence in making decisions about the care of individual patients. The aim of proof-based medicine is to integrate the experience of the clinician, the values of the patient, and the best available scientific information to guide decision-making about clinical management. In 5P medicine, the medical decision is made by considering the three parameters cited above. In fact, the clinician must consult the scientific literature or guidelines (domain specific knowledge) to solve clinical problems and offer the optimal decision to the patient [195]. Doctors can also use their clinical experience which is based on a systematic

analysis of clinical observations avoiding any intuitive interpretation of the information. The Patients preferences should be considered in making clinical decisions about their care.
To meet the requirements of 5P medicine, clinicians use Decision Support Systems. According to Berner, Clinical Decision System (CDS) [29] provide to clinicians, staff and patients information's intelligently filtered and presented at appropriate times, to enhance health and health care.
CDS can assist physicians by offering them a summary in their daily practice, if this really meets their expectations.
Despite the many achievements that have emerged over the past twenty years, the most of these systems suffer from a lack of explanation ("interpretability ") and transparency since there is no explanation behind the provided decisions.
To overcome this shortcoming, it is necessary to add transparency to CDS to justify the decisions and allow the clinicians interacting with the system.

In this thesis, we propose a new approach for adding transparency to the CDS. Such an approach gives the CDS the ability to:
⇒ Provide decisions with explanation;
⇒ Give the clinicians the possibility to control the decision-making process by injecting prior knowledge and/or eliminating conflicting ones.

The fact of involving the expert in the system avoids contradictory results that can be uninterpretable. In fact, injecting prior knowledge in CDS is desirable to guide the decision process and to reduce their lack of explanation.
In this thesis, we are specifically focused on Hypertension. According to data validated by the French Committee for the Fight against Hypertension established in 2017 in France, Hypertension affects more than 13 million people [87] and 1 billion individuals worldwide [136] [28]. It can cause very serious complications like heart attacks, strokes, etc.
Thus, the individualization of the follow-up can be used in the context of Hypertension to detect and diagnose diseases in a highly specific manner. This allows treatments which are increasingly specific and effective, improving the prognosis of many.
To predict the public health impact cited above, we need human intelligence. But to perform studies on large amounts of data and population-level predictions, we constantly need to use AI based systems. Therefore, certain disciplines must be added to achieve the goal of Explainable 5P medicine.
First, medical measurements need to be collected from each patient to build Electronic Health Records (EHR). EHRs is then used for analysis with AI based algorithms. This enables a deeper interpretation of an individual's characteristics and a disease mechanism understanding, which will lead to better treatment and prevention. Otherwise, it is about allowing the person to take care of himself/herself and to have access to information and advice adapted to his/her profile.

In this thesis, we include domain specific knowledge into the CDS by using two areas of AI: Machine Learning (ML) and Multiagent system (MAS).

1. ML offers a wide range of algorithms based on a large amount of data which can reproduce a behavior. Faced with many situations, the algorithm learns which decision to make and creates a model. The machine can automate the tasks according to the situations. In addition, it can exploit the volume of the EHRs.

2. MAS provide the possibility of effectively representing the different kinds of data and knowledge allowing to deal with the CDS complexity. Moreover, the integration of argumentation for modelling agents interactions adds transparency and intelligence to the system.

1.2 Context

In this thesis, we are interested in Hypertension as a cardiovascular disease. Nowadays, Hypertension is a major cause of premature death worldwide [59] [93].
Hypertension, also known as high or raised blood pressure, is a condition in which the blood vessels have persistently a raised pressure. Blood is carried from the heart to all parts of the body in the

vessels. Each time the heart beats, it pumps blood into the vessels. Blood pressure is created by the force of blood pushing against the walls of blood vessels (arteries) as it is pumped by the heart. The higher the pressure, the harder the heart must pump. Hypertension is usually defined by a blood pressure of 140/90 mmHg or higher for auscultatory measurement (mercury or aneroid) performed in a standardized manner. For patients with diabetes, high blood pressure is defined as blood pressure of 130/80 mmHg or higher.
Antihypertensive therapy is designed to reduce blood pressure levels below 140/90 mmHg to minimize the risk of cardiovascular complications in the long term. Several factors favoring Hypertension can be modified by simple lifestyle and dietary measures [272] [111]. In particular, the practice of moderate physical activity for at least 30 minutes a day, a moderation of salt intake and alcohol consumption, and lastly, loss weight if necessary are recommended. These recommendations are restrictive for patients because they may impose a change in their lifestyle, but they are effective if they are put into action seriously. Nevertheless, in the absence of improvement after few months, a hypotensive treatment becomes necessary. It will often be maintained for the whole life if it can effectively control the blood pressure. There are several therapeutic classes. Some of them can be combined for a cumulative effect [153] [104] [17]:

1. Thiazide diuretics act on the kidneys and promote the elimination of water and salt;
2. Beta-blockers inhibit the stimulating effect of adrenaline on the heart and slow down the heart rate, thus limiting the intensity of blood pressure on the artery walls;
3. Calcium inhibitors slow down the entry of calcium into the muscle cells of the arteries, causing their vasodilation and thus a drop in blood pressure;
4. ACE inhibitors and angiotensin II receptor blockers (ARA2) both block the renin angiotensin system involved in the blood pressure level;
5. Alpha receptor inhibitors act on the alpha1 receptors of cells that make up the wall of blood vessels. They are most often prescribed in case of failure of at least two other treatments.

One of the problems of Hypertension is related to its late diagnosis because of different factors such as those defined bellow:

- White coat hypertension [270] [76], also known as white coat syndrome, is a condition where a patient's blood pressure is higher when taken in a medical environment than it is in other environments, such as at home. The term received its name from the white coats that medical professionals wear;
- Masked Hypertension [36]: is the opposite of white coat Hypertension. Patients with masked hypertension have normal blood pressure readings at the doctor's office but have high blood pressure readings in other settings, such as in their home;
- Anxiety [201]: while occasional anxiety does not cause long-term high blood pressure, it can cause temporary spikes in blood pressure during those anxiety episodes. If those temporary spikes in blood pressure occur every day or frequently, they can damage blood vessels, the heart, and kidneys in the long run or cause high blood pressure.

Hypertension varies throughout the day. It is lower during sleeping and resting and higher otherwise. In addition, it increases under the effect of several parameters: physical activity, cold weather, emotional shock, stress ...
All these concerning conditions might lead to a wrong diagnosis, medication errors or even a late diagnosis due to the absence of symptoms.
Thus, the need of frequent blood pressure monitoring becomes a necessity. The diagnosis of hypertension must therefore be confirmed by repeated measurements. Collecting repeated medical measurements and high-quality health data for the patients and connecting them for analysis with automatic and intelligent tools avoids misdiagnosis and/or overmedication. A population of individuals should be used to characterize patients' profiles. Then, the idea is to progress these profiles to refine the individual management advice. This enables to understand the diseases mechanism, which will lead to better treatment and prevention.
In this thesis, we will study tools that we consider the best to produce personalization in medicine and treat a large amount of health data with an emphasis on transparency and interpretability.

1.3 Challenges

In this section, we cover the main challenges that should be considered in machine learning systems and multiagent systems for healthcare tasks. We distinguish two types of challenges: applicative and scientific challenges.

1.3.1 Applicative challenges

The extent and increasing complexity of medical knowledge, such as diagnostic and therapeutic tools, oblige clinicians to manage more and more information to treat a patient. When treating a patient, the doctor must make a whole series of decisions leading to the medical act. The Doctor acts by following a reasoning which simultaneously involves notions of knowledge, uncertainty, experience, and risk. The development of CDS, simulating medical reasoning, requires modeling this practice. For this purpose, it is important to retrace the doctor's approach to a patient and to analyze the medical decision. For designing a CDS, medical history of the patient (diagnosis, treatment, allergies, etc.) is required. Such data is called EHR [35]. An EHR is a digital version of a patient health information. A wide range of information can be stored in EHRs, such as detailed records of symptoms, data from monitoring devices, clinicians' observations, radio, laboratories tests... A typical EHR consist of heterogeneous data elements, including patient demographic information, diagnoses, laboratory test results, medication prescriptions, clinical notes, and medical images. EHRs, even in their simple form, provide a rich collection of data for the researchers. As the number, the volume and the resolution of temporal datasets increase rapidly, traditional methods for dealing with such data are becoming overwhelmed. Therefore, one of the major challenges consists in integrating and harmonizing data belonging to different institutions. In addition, the heterogeneous nature of the data types including numerical data, date, time, objects, free text, images etc. poses significant challenges in working with EHRs. Mixed EHR type data drives an interesting research field of how to treat and to combine them for learning prediction models. Bellow, we specify some key challenges in modelling structured EHR data for ML research:

- Heterogeneity [114]: EHRs hold a various type of data including symptoms, demographics, procedures, diagnoses, lab exams and prescribed medications. The data is heterogeneous both in terms of clinical concept and data type. For instance, date time objects such as visit dates and time; multi-range numerical quantities such as lab results; categorical data such as diagnostic codes or visit locations;

- Dimensionality [149]: another inherent challenge with EHR data is the dimensionality of clinical concepts. There exist thousands of different procedures, labs, and medications in medicine. The problem of high dimensionality increases the model complexity by adding more parameters and the number of training samples in order to avoid model overfitting;

- Temporality [292]: EHRs include time-stamped sequences of measurements (clinical visits) over time which contain important information about the progression of disease and patient trajectory over the care period. The sequences are irregularly sampled. Both the order of clinical events and the duration between events are valuable pieces of information for learning prediction models.

These factors might affect the performances of different prediction models. So, the first goal is to find a way to pretreat EHRs in order to predict the treatment of Hypertension and the date of the next visit. It is a question of surely estimating the treatment to allow patients taking care of themselves and especially to be able to anticipate and detect situations where it is essential to go see a doctor.

1.3.2 Scientific challenges

Traditional methods are not performing well enough to fully exploit the value of EHRs. The volume of data is too large for comprehensive analyzes, and the relationships between information contained in this data are too large for analysts to test all hypotheses to derive value from the data. ML is ideal for exploiting the hidden opportunities of EHRs. This technology allows to extract value from massive and varied data sources without the need to rely on a human. It is purely data-driven and suited to the complexity of the immense data sources. Unlike traditional methods, ML algorithms can be

applied to growing data sets. In fact, ML algorithms generally need a large amount of data for better performance. Thus, they deal with EHRs since the massive and varied data used in a ML system can improve learning results. ML allows discovering patterns buried in the data with greater efficiency than human intelligence.
Efficient ML algorithms are certainly very effective. However, they acquire some disadvantages. Bellow, we specify some key challenges when using some efficient ML models:

- Lack of interpretability: this is the case with neural networks [79], which are extremely powerful for prediction, but whose operations remain mysterious. However, we find these algorithms in uses where the slightest error can be fatal such as models used for healthcare. How to make a ML algorithm interpretable or explainable? A real challenge when it comes to enlightening a decision-maker on the result of an AI that he/she is supposed to use to inform his/her operational or strategic orientations. How could he/she decide if he/she does not understand how the model works, if he/she has no way of appreciating the logical process that led to the generation of this or that indicator? Transparency also rhymes with confidence, and especially when we talk about ML as a new step in the automation of systems;

- No ability to integrate prior knowledge [285]: ML models are obtained from training large amount of data. This purely data-driven learning may induce contradictory results that can be uninterpretable. Injecting prior knowledge in the leaning model can guide the learning step of the models and reduce their non-interpretability. Since we are working in the medical domain, we focus on official recommendations which are provided by official European Society of Cardiovascular Diseases and clinical knowledge;

- No ability to exploit internal ML knowledge: ML models only integrate classification results rather than internal classification knowledge.

1.4 Global vision of our support decision system

In this thesis, we propose an original vision to design a CDS for the prediction of the optimal treatment and the date of the next visit. Otherwise, we propose an original method, based on multiagent argumentation, which combines several Deep Learning (DL) algorithms which are one of the most powerful ML algorithms [162]. This way of combining DL algorithms allows not only to provide explanations of individual predictions, but also to inject prior knowledge and exploit the internal knowledge of each classifier. Thus, the argumentation process uses knowledge extracted from the individual built models and prior knowledge.
Our method proceeds in three main phases (see Figure 1.1): (1) Machine learning phase, (2)Arguments construction phase and (3) Multiagent argumentation phase.

1.4.1 Machine learning phase

Rather than making one model and hoping this model is the best/most accurate predictor, we use several models, first step, to construct ensemble method which improves ML results by combining different models. After processing data, each ML model generates a sample by using a specific samlping technique such as bootstrap. ML models use this samples to learn, to discover patterns and to make predictions. This phase is very important in our system since it allows to build knowledge bases. In this phase, the choice of the ML model, which represents the basic building block of our architecture.

1.4.2 Arguments construction phase

Arguments construction phase consists in building arguments either from machine learning algorithms or from prior knowledge. Extracting knowledge from different classifiers allows to make the link between ML and the multiagent system. The idea is to build an interpretable model that imitate the behavior of DL [118]. We opted in this work to extract if-then rules because this is one of best way to explain the prediction for clinicians. Prior knowledge is used under if-then rules form and injected into the system to guide the decision process.

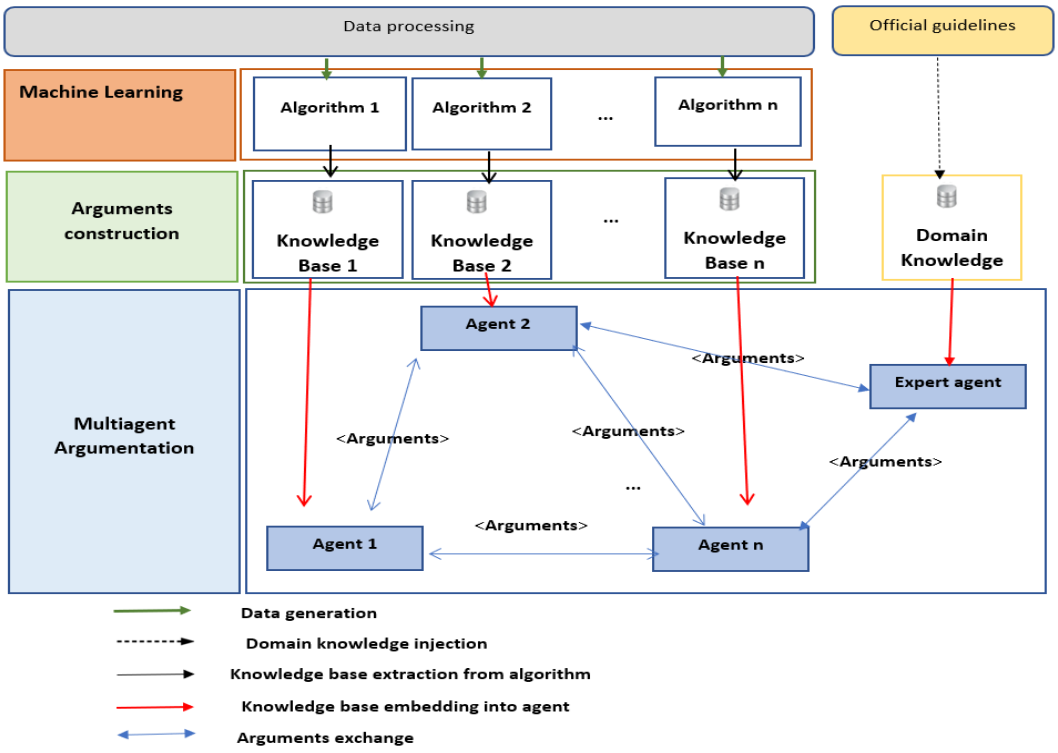

Figure 1.1: General approach architecture

1.4.3 Multiagent argumentation phase

Ensemble methods require a mechanism to combine base classifiers for generating ensemble classifier. Therefore, in this process, we introduce argumentation technology to integrate base classifiers in the MAS and to make the classifiers interact and exchange arguments for prediction. This new ensemble strategy based on multiagent argumentation can overcome the weaknesses of ML methods cited in the previous subsection. Argumentation technology [217], as a new way of multiagent interaction, can imitate human decision-making process to realize the conflict resolution and knowledge integration, which takes advantage of collective intelligence for problem solving. In this phase, extracted and prior knowledge are embedded into the agents. These agents negotiate in order to provide the most relevant prediction.

1.5 Plan

To present our work, we have organized this manuscript into three parts:

1. Part 1
   In the first part, we set out the background of the involved domains and the state of the art as following:

   - Chapter 2 provides an overview of the most widely involved areas: machine learning and multiagent argumentation. This Chapter presents the background which is crucial for the understanding of the thesis. First, we present some ML algorithms, a crucial step to choose the most efficient model for our problem. Then, the second part of this chapter is devoted to argumentation theory and multiagent system;

   - Chapter 3 establishes a state of the art based on the two fields cited above. We present the approaches which are related to our project in the literature either in machine learning, multiagent systems or argumentation in healthcare. We raised the issues to be resolved in each part.

2. Part 2
   The second part of this thesis encompasses the contributions of this thesis by describing them through two chapters as following:

   - Chapiter 4 presents the proposed medical support system that we have designed as well as the preliminary results allowing to validate our approach;
   - Chapiter 5 presents a new model able to treat times series called MS-LSTM. The idea is to replace the classical ML algorithm used in ML phase of our medical support system to consider the temporal trajectory of EHR. The proposed model can combine different data sources and predict clinical events. In addition, it allows to understand what group of features contributed to the prediction.

3. Part 3
   Finally, the last chapter of this thesis is the subject of a general conclusion where we summarize the different obtained results and highlight the contributions of this work. We also describe the open perspectives and new research directions to explore.

Chapter 2

Background

Contents

2.1 Introduction

In this part, we will focus on presenting the two areas on which our approach is based: ML and multi-agent argumentation. Both have specific interests for the problem we are addressing, and it is therefore essential to be able to provide the reader with the keys to understand the work that will be presented later. For this purpose, the foundations, objectives, and classic challenges will be discussed.

2.2 Machine learning

As a subfield of AI, ML is a branch born out of the desire to train machines to perform tasks typically assigned to human beings. We can cite for example the recognition of numbers or letters. Concretely, this approach consists in making the machine learn concepts or rules from examples of data sets. Once this step is performed, we can then give new examples to the machine so that it can predict their outputs. What is sought during this process is the development of the machine ability to generalize through the learned concepts. This way of operating makes it possible to solve complex problems which are known to be too costly in terms of time and resources by the traditional algorithms. ML has been a hot topic in recent years [184] and its use is becoming more widespread in many fields such as the stock market, robotics, image recognition, medicine etc. [126]. One of the factors behind this success is the ability of the ML algorithms to match or even surpass human performance.
ML algorithms can be subdivided into four categories depending on the type of data they are facing and the task to reach. The first category is called Supervised learning [143], where the input data used for the learning process has a label indicating the output value that would be expected during a prediction. The chosen algorithm therefore has information on the concepts it needs to learn. Supervised learning attempts to solve two problems: Classification and Regression.
The second type is called Unsupervised learning [80]. The aim of this kind of learning is to see what it can be learned from the data without labels. The model will have to deduce by itself the relevant criteria and concepts, grouping examples that seem to correspond to the same profile or reducing the dimension of data. This type of ML is typically used to discover structures and patterns in data. It can also be used for Feature engineering during the process of preparing data for supervised learning.
The third type is the Semi-supervised learning [21]. It consists in working with data with and without labels. Overall, it is observed that the association of the two categories of data tends to improve the accuracy of the models. An example of semi-supervised learning is co-learning, in which two classifiers learn a set of data, but each using a set of different and independent characteristics. If the data are individuals to be classified into men and women, one can use height and the other hair growth for example.
Finally, the fourth type is called Reinforcement learning. An agent starts by choosing an action from a list of actions. Then, depending on the chosen action, it will receive a feedback from the environment (coming from a human in certain situations or implemented inside the algorithm): it is either a reward for a good choice, or a penalty for a bad deed. The algorithm learns which strategy (or choice of actions) maximizes the stack of rewards. This type of learning is often used in robotics, game theory and autonomous vehicles.
In this thesis, the task is to predict treatment or date of the next visit for a patient. It is necessary here to have labels to have a scale on which the machine can rely for its predictions.
We thus focus on the supervised ML algorithms by dealing with classification and regression problems. The classification encompasses all the problems in which we will seek to predict a class, a category. It is possible to perform classification in a supervised manner. Like classification, regression needs labels. Regression consists in finding a model or a function to distinguish data in continuous real values instead of using discrete value. The significant difference between classification and regression is that classification assigns the object of the input data to certain discrete labels. On the other hand, regression affects the input data object to actual continuous values. Simply speaking, in classification, we try to answer the question "which class?" however with regression, we attempt to answer the question "how much? ".
In this thesis, we attempt to answer two questions: "which treatment" which is considered as a classification problem and "how soon will the patient have a next visit" which refers to a regression problem or a classification problem if we discretize the duration between visits.
To evaluate the model for a classification problem, different evaluation metrics could be used such as

Accuracy, Precision, Recall and F1-Measure which calculate the performances of the model. These metrics are described above, where TP, FP and FN, TN represents respectively the number of true positive, false positive, false negative and true negative examples:

- Accuracy: it indicates the percentage of well-predicted data;

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{2.1}$$

- Precision: it measures the accuracy of positive examples;

$$Precision = \frac{TP}{TP + FP} \tag{2.2}$$

- Recall: it measures the number of positive labels well ranked among all the positive labels;

$$Recall = \frac{TP}{TP + FN} \tag{2.3}$$

- F1-Measure: it conveys the balance between the precision and recall.

$$F1 - Measure = \frac{2 \cdot TP}{2 \cdot TP + FP + FN} \tag{2.4}$$

Regression results are evaluated differently than classification. One of the commonly used metric is the Root Mean Squared Error (RMSE). RMSE is essentially the square root of the sum of the squared differences between the prediction ($\hat{y}$) and the expected (y):

$$RMSE = \sqrt{\frac{1}{n}\Sigma_{i=1}^{n}\left(\hat{y}_i - y_i\right)^2} \tag{2.5}$$

To fill out the objectives cited previously, data is crucial for learning, testing, validating, and monitoring the ML algorithms. In this thesis, we use a specific type of data which is EHRs of patients. In the context of EHRs, the temporal aspect is present at different levels of granularity (sequence of visits, diagnosis, etc.). In the case of such temporal data, each event depends, in most cases, on the previous events. The temporal aspect is important because there are many prediction problems that involve a time component. While the time component adds information, it also makes time series problems more difficult to handle compared to other prediction tasks because of their irregularity. Some approaches are more suitable than others for exploiting time series. In this thesis, we grouped ML algorithms into two main approaches:

- Classical learning methods: include models that do not take into consideration the relationship between the events and the temporal aspect inside the algorithm;
- ML algorithms for time series: methods that can model sequential and temporal data inside the algorithm.

Based on the above, we organize the rest of this chapter as follows:
We present, in part 1, some classical algorithms. In part 2, we present some ML methods for time series and we discuss their ability to consider temporal aspect of EHRs. Then, we discuss in part 3, different techniques of interpretation of DL algorithms.

2.2.1 Classical learning algorithms

In this section, we present some of the classical learning algorithms which include Shallow ML and DL algorithms. We also expose the advantages and the weaknesses of these algorithms.

Shallow Machine Learning algorithms

1. Decision tree

   The Decision Tree (DT) [211] is a supervised learning technique that can be used for regression or classification problems. The construction of a decision tree begins by selecting recursively the attribute that most effectively separates the learning data. The first selection thus provides the attribute of the root of the tree accompanied by the conditions (branches) relating to its value. Then, the child node attached to each branch is chosen either as a leaf of the tree, and therefore a class, or as an attribute developing a sub-tree in the same way as the root node.
   The decision tree is very easy to interpret. For example, from the tree shown in figure 2.1, we can deduce the following two rules:

   - If the patient has a difficulty breathing or a bleeding wound then he/she will be sended to the hospital;
   - If the patient does not have a difficulty breathing or a bleeding wound then he/she will stay at home.

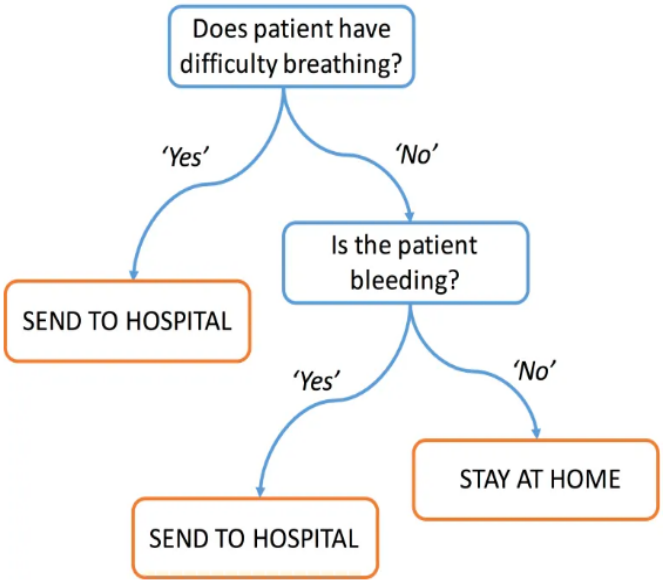

Figure 2.1: Decision tree for deciding if a patient will be hospitalised or stay at home [233].

From the above, we notice that the selection of the optimal attribute according to which the data will be distributed, is a fundamental step. Several methods have been proposed to find the optimal attribute such as information gain and the Gini index [218]. On the other hand, several studies have shown that there is no optimal method. Various algorithms for constructing decision trees have been developed, such as the CART [44], SLIQ [177] and SPRINT [246] and C4.5 algorithm [212].
One of the advantages of DTs is the possibility for the user to understand, through the routing throughout the conditions relating to the attributes, the reason why the model assigns a particular class to an input vector. However, given its individual concentration on each of the attributes, DTs algorithms are not very suitable for processing temporal data. In addition, DTs are particularly intolerant to missing data problems.

2. k-Nearest Neighbors

   The k-Nearest Neighbors (kNN) algorithm belongs to "lazy" learning algorithms. With each new instance, kNN is based directly on the instances of the training data without building a model. The assumption on which the kNN algorithm is based is that an instance is closer to instances of the same class than those of other classes. Therefore, when classifying an unknown

instance, the algorithm looks at the most frequent class among the classes of the K closest instances according to a defined distance. An example of classification by kNN is presented in figure 2.2. In this diagram, the first class is represented by a blue circle and the second by a red tile. The new instance to be classified is in the form of a cross. Using a 3NN classifier (KNN with $k = 3$), all the 3 closest neighbors of the new instance belong to the first class. The algorithm then considers that this instance belongs to the first class.

The choice of the value of k influences the performance of the kNN classifier. In the example

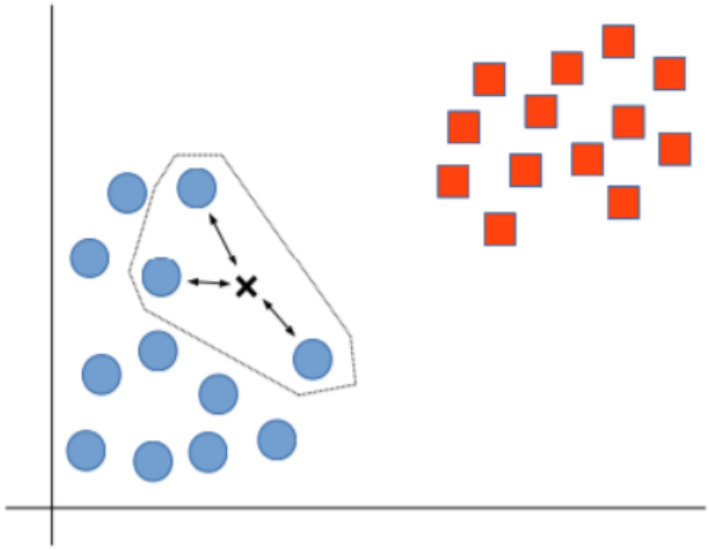

Figure 2.2: An example of classification by 3NN.

2.3 (a), choosing a too small value made the system sensitive to the noise present in the first class area (represented by circles). Among the neighbors of the instance to be classified, the 3NN algorithm finds 2 instances of the second class against one belonging to the first class. This new instance was therefore wrongly considered to belong to the second class. This has led to a state of overfitting. In this case, a larger value of k ($\geq$5) can solve the problem. On the other hand, in example (b) the choice of a large value led to overfitting. Given the small number of instances present in the central region, the 8NN classifier is still looking for neighbors in the region of the other class. This problem can be solved by taking a smaller value of k ($\leq$ 5).

After fixing the number of neighbors k, kNN uses widely Euclidean distance to detect the

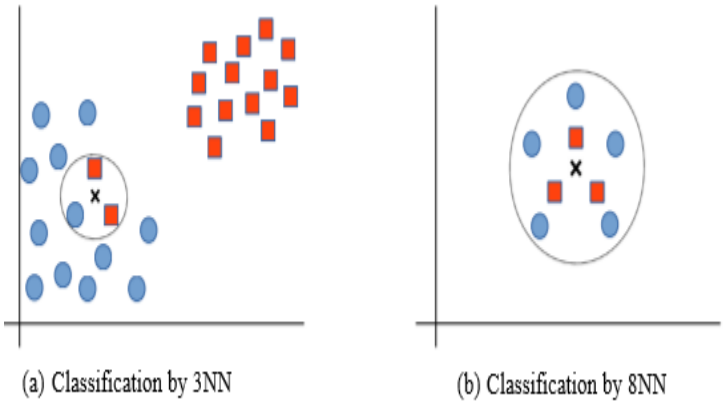

Figure 2.3: Importance of the choice of the value k in the classification by kNN.

k-neighbors closest to the new input data to be classified.

3. Support Vector Machines

Support Vector Machines (SVM) [256] are supervised learning algorithms. Although there are versions able to solve regression problems, their application is more common for classification problems. SVMs can solve linear and non-linear problems and work well for many practical problems:

(a) Linear Support Vector Machines

Linear SVMs are the simplest form of this algorithm. They are applicable in the case where the data are linearly separable. Linear separation consists in finding an optimal hyperplane which separates data into two classes. Since there are an infinity of possible hyperplanes, the quality of the separator is evaluated by maximizing the margins. The margins are calculated according to the distance separating the hyperplane from the closest element belonging to each class. When the margins are larger, the separation is better between the classes.
Given a dataset of N instances (x_i, y_i), $i = 1...N$ where $y_i \in \{-1, 1\}$ represents the class of x_i. The goal is to build a function f which allows to predict whether a new example x_i belongs to class -1 or to class 1. For a linear classification problem, the two classes (-1 and 1) are supposed to be separable by a hyperplane, the function f therefore has the form:

$$f(x) = w^T x_i + b = 0 \tag{2.6}$$

where w is the orthogonal vector to the hyperplane and b is the displacement from the origin.
And satisfying the following constraints:

$$\begin{cases} w^T x + b \geq 1 & if\ y_i = 1 \\ w^T x + b \leqslant -1 & if\ y_i = -1 \end{cases} \tag{2.7}$$

The "margin" of a learning problem is defined as the distance between the closest learning example and the separation hyperplane. The distance between the two hyperplanes ($w^T x + b = 1$) and ($w^T x + b = -1$) is $\left\| \frac{2}{w} \right\|$ and represents the margin of the classifier. The optimal hyperplane can be found by maximizing the margin, or equivalently by solving the following minimization problem:

$$\min \tfrac{1}{2} \|w^2\|$$

Under the constraint:

$$y_i(w^T x_i + b) \geq 1, \forall i \leq N \tag{2.8}$$

Figure 2.4 shows an example of SVM separation of two classes 1 and -1.

(b) Non-linear SVM

Unlike Linear SVM, which deals with the data where two classes are linearly separable, Non-linear SVM can handle with data where the classes are not linearly separable. A Non-Linear Kernel [68], like for example the RBF kernel (Radial Basis Function Kernel) [239] can be used in order to transform the non-linear data into almost linearly separable data. More precisely, a kernel function [5] is applied on each data instance to map the original points into some higher dimensional space in which they become linearly separable.
The process of determining the classification function in this case consists of two steps:

- First, the input vectors are projected into a larger dimensional space so that they can be linearly separable;

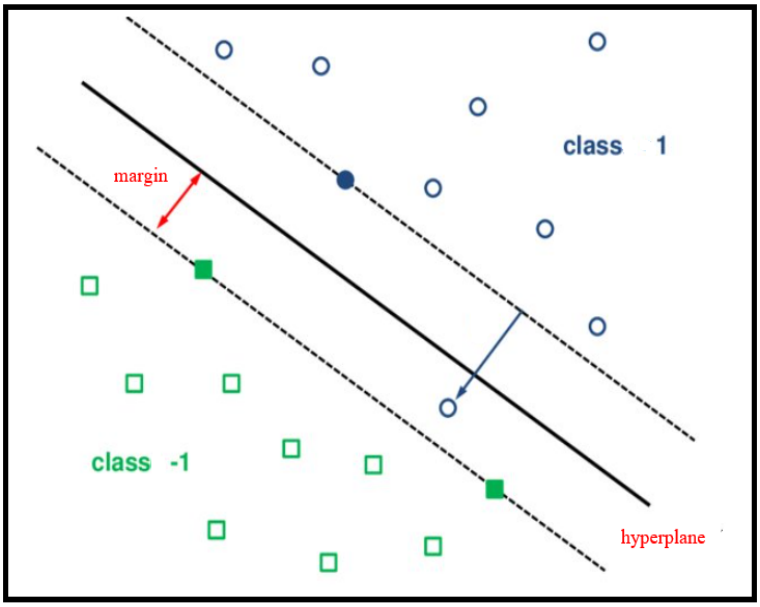

Figure 2.4: SVM classification with maximization of the separation margin [267].

- Then, the SVM algorithm is used to find the optimal hyperplane that separates the new data vectors. This hyperplane is therefore defined by a linear function in the new space that corresponds to a non-linear function in the original space.

Let ϕ be the function of projection of the data in the destination space. After this projection, a learning algorithm could manipulate the data through a scalar product in this destination space.
Kernel functions are special functions that allow to compute scalar products directly in the original space without the need of determining the projection ϕ. A kernel function $K(x_i, x_j)$ represents the similarity between x_i and x_j which is given by:

$$K(x_i, x_j) = < \phi(x_i) \cdot \phi(x_j) > \tag{2.9}$$

Among the most used kernels, we can cite polynomial kernels and RBF kernels formulated respectively as follows:

$$K_{poly}(x, \tilde{x}) = (x^T \tilde{x} + c)^d \tag{2.10}$$

$$K_{RBF}(x, \tilde{x}) = \exp(\frac{\|x - \tilde{x}\|^2}{2\sigma^2}) \tag{2.11}$$

SVMs produced good results in ML tasks, mainly in classification and regression; such results have been observed on several problems.

4. Ensemble methods
Ensemble methods [81] are based on the idea of combining the predictions of several classifiers for better prediction results. Depending on how to generate the components of the classifiers, current ML algorithms fall into two categories: (a) algorithms which generate components of classifiers independently [230] (Bagging, Random Forest,..) and (b) algorithms which generate components of classifiers sequentially [86] (Boosting, Arcing,...). Here, we detail one of each category as follows:

(a) Bagging
Bagging is an ensemble method introduced by Breiman [45]. The concept of bagging (voting for classifications, averaging for regression problems with continuous dependent variables) finds its application in the field of predictive ML, to combine the classifications predicted from the same type of model for different training data. It is also used to solve the inherent problem of instability of results when complex models are applied to relatively small data sets.

Bagging is based on the concepts of Bootstrapping and aggregating. Bootstrap [262] is a principle of statistical resampling [262] traditionally used for the estimation of quantities or statistical properties. Bootstrapping is designed to generate randomly and with delivery N independent copies of S objects called bootstrap from the initial set of training samples with size S. An object from the initial dataset can be selected multiple times as it can be absent in generated copies. The same classifier is learned on each of the copies. N classifiers are then obtained with different performances.
Aggregation consists in combining these classifiers using majority voting as the combination strategy. The final classification will be the one predicted by the greatest number of classifiers.
For a regression problem, the outputs of individual models can literally be averaged to obtain the output of the ensemble model.

(b) Boosting
Boosting is an ensemble method introduced by Schapire [238]. Their goal is to improve the prediction results and get a better classifier from a poorly performing classifier. By successive iterations, the knowledge of a weak classifier is added to the final classifier.
Unlike bagging where bootstrap training sets and classifiers are built independently, in boosting training samples are built incrementally by the same classifier and sequentially. Initially, all training samples have equal weights, and the classifier is built on this basis. Then for each step, the samples are weighted so that the misclassified objects have high weights, and the classifier is launched on the new set of learning weights. In this way, we finally obtain a set of classifiers which are combined by a weighted vote to have the final decision. These methods can be applied to regression as well as classification problems.

Despite their advantages, ensemble methods have several limitations. In fact, they proceed by a simple voting or averaging, they do not exploit internal knowledge of classifiers, they do not allow knowledge injection and they do not provide an explanation for the prediction.

Deep Learning

In recent years, DL that represents a new generation of neural networks have proven their effectiveness in many real applications. It is particularly promising in the fields of images, sound recognition, natural language processing, games, and medicine. As their name suggests, neural networks are conceptually inspired by the functioning of the human brain and the neurons. This network is made up of “layers” of neurons, each receiving and interpreting information from the neighboring layer. Bellow, we present some basic neural networks.

(a) Formal neuron

The architecture of artificial neural networks brings together a set of elementary units called "formal neurons". These neurons are connected to each other to form an oriented graph. In analogy with the biological networks of neurons, the connections between the nodes of the graph symbolize the synapses. These connections are weighted by adjusted the weights during the learning phase by means of an optimization algorithm. This type of algorithms adapts the weights to minimize the difference between the network output (the predictions) and the expected output (the reference). The behavior of the neurons in the nervous system was used to create the mathematical concept of "formal neurons". These neurons receive information produced by other nodes through the input connections. The artificial neuron first performs a weighted sum of the n input values $x_1, x_2 ... x_n$. The weights assigned to the inputs of a neuron are stored in a vector W, where the value w_i represents the weight of the input connection x_i of the neuron. To this sum is added the threshold value b which represents the "bias". This total quantity represents the biased

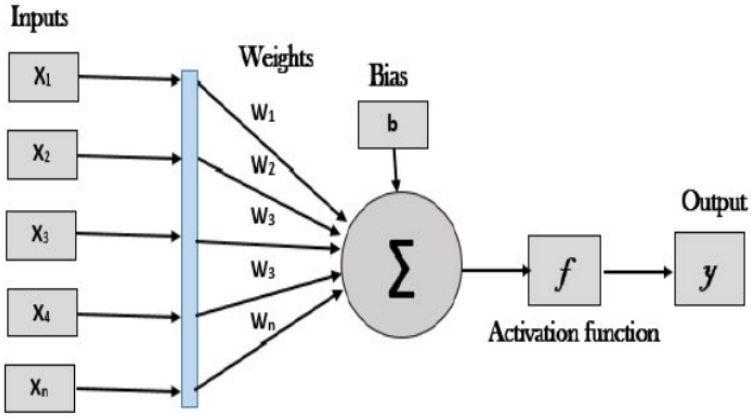

Figure 2.5: Representation of a neuron [95].

post-synaptic potential p formulated as follows:

$$p = \sum_{i=1}^{n} w_i x_i + b \tag{2.12}$$

Finally, an activation function f transforms this biased potential to obtain the activation value of the neuron which can then be transmitted to other neurons (see figure 2.5).

$$y = f(p) \tag{2.13}$$

Among the commonly used activation functions, we can cite Relu and the sigmoid functions:

$$Sigmoid \rightarrow f(x) = \frac{1}{1 + e^{-x}} \tag{2.14}$$

$$Relu \rightarrow f(x) = \begin{cases} x & if\ x \geq 0 \\ 0 & if\ x < 0 \end{cases} \tag{2.15}$$

The perceptron is a very simple unit. However, combining many perceptrons forms an artificial neural network which can theoretically solve complex and undefined problems like humans do. Amoung neural networks, we find Multi layer perceptron.

(b) Multi-layer perceptron

Multilayer Perceptrons (MLPs) [220], shown in Figure 2.6, are non-looped neural networks whose nodes are organized into three or more levels called "layers". The neighboring layers are completely connected, which means that the nodes of each layer are linked to all the nodes of the lower layer and to all those of the upper layer. However, no connection exists between the units of the same layer. An MLP is made up of three types of layer, an input layer, one or more hidden layers, and an output layer:

- Input layer: is the first layer of the network. Neurons of this layer receive the information provided by the input vectors of instances. This layer therefore has no input connections from other nodes. It is however completely connected to the first hidden layer.
- Hidden layers: given a MLP containing N ($N \geq 1$) hidden layers, each of the $N - 1$ lower hidden layers are completely connected to the upper one. The N-th and last hidden layer is completely connected to the output layer.
- Output layer: in this layer, the output of the final perceptrons is the final prediction of the perceptron learning model.

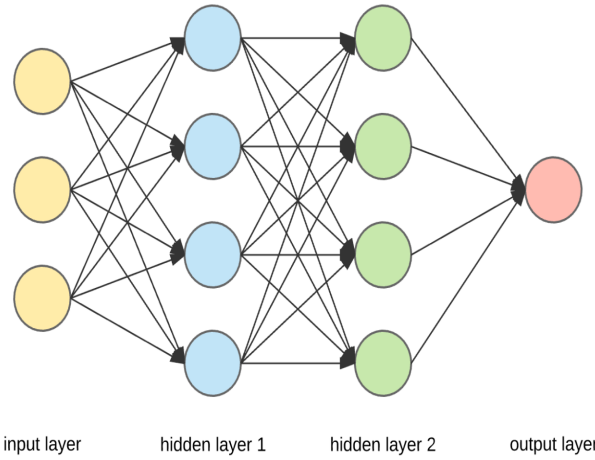

Figure 2.6: Representation of a neural network with two hidden layers [150].

The MLP learning process consists in adapting the connection weights according to the prediction errors observed for each classification of a new instance. It starts by the feed-forwarding the information from one layer to the next. This goes through two steps that happen at every node/unit in the network:

i. Capturing the weighted sum of inputs of each neuron;

ii. Applying an activation function on the output of the step (i). Obtained activation value (i.e. the output of the activation function) is the input for the connected neuron of the next layer and so on.

At the end of this process, we may have a model that does not give accurate predictions and that is attributed to the fact that its weights have not been tuned yet. We also may have a high error. Backpropagation [220] is the most widely used method for adjusting the weights. This algorithm determines the error gradient for each neuron in the network starting from the last layer to the first hidden layer. During training, at each epoch, there will be a backpropagation of the error. The weights associated with the neurons are then modified to best rectify the error in the next step.
The objective of the backpropagation of the gradient is to adjust the weights of the connections to minimize the error, as an example, we can use quadratic error:

$$E = \frac{1}{2}\sum_{i=1}^{N}(y_i - \tilde{y_i})^2 \tag{2.16}$$

which represents the difference between the expected output (the labels, y) and the output produced by the network (predictions, $\tilde{y}$). N represents the size of the training dataset. The backpropagation is performed by using an optimization algorithm (Gradient Descent, stochastic gradient descent, ADAM ...). The algorithm will find the weights that will hopefully yield a smaller error. Considering that the error E is a function of the weights w, a local minimum is targeted by changing the weights in the opposite direction to the gradient $\frac{\partial E}{\partial w}$ multiplied by the learning rate α :

$$\Delta w_{ij}^{*} = -\alpha\frac{\partial E}{\partial w} \tag{2.17}$$

$$w_{ij} = w_{ij} + \Delta w_{ij}^{*} \tag{2.18}$$

A chain of rules is used for the calculation of the gradient. For more details, authors in [185] provide a detailed overview.

(c) Auto-encoders

Autoencoder (AE) [191] is an unsupervised learning algorithm. The goal of an autoencoder is to learn a new representation of a set of data. Thus, the network may be viewed

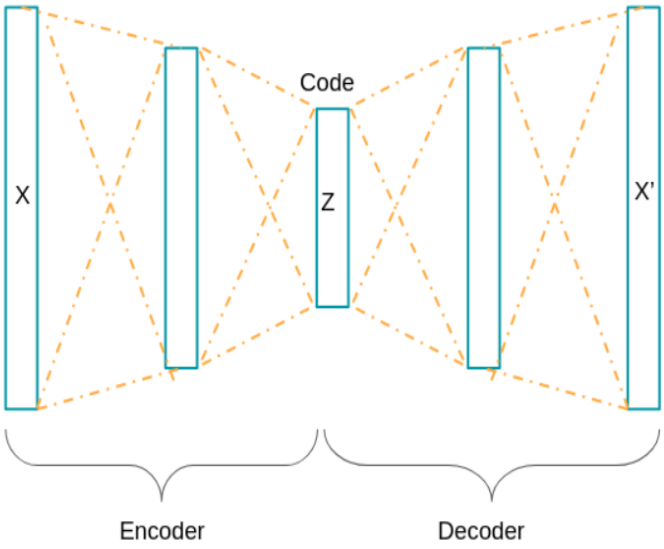

Figure 2.7: Representation of an autoencoder [110].

as consisting of two parts: an encoder function $Z = f(X)$ and a decoder that produces a reconstruction of the inputs $X' = g(Z)$. This architecture is presented in figure 2.7. If an autoencoder succeeds in simply learning to set $g(f(X)) = X$ everywhere, then it is not especially useful. Instead, autoencoders are designed to be unable to learn to copy perfectly. Usually they are restricted in ways that allow them to copy only approximately, and to copy only input that resembles the training data. Because the model is forced to prioritize the aspects of the input to be copied, it often learns useful properties of the data. The network will therefore represent the data by means of one or more hidden layers so that the output finds the same data as the input. An autoencoder can discover structure within data in order to develop a compressed representation of the input. Because autoencoders learn how to compress the data based on attributes discovered from data during training, these models are typically only capable of reconstructing data similar to the class of observations that the model observed during the training phase.
Autoencoders are mainly applied for dimension reduction, anomaly detection, data denoising (ex. images, audio), image inpainting, etc.

(d) Convolutional Neural Network

The visual cortex of animals is endowed with several cells each having their responsibilities, some for detecting light, others for contrasts... These cells act like a series of filters that only lets one image pass to the animal's brain. Convolutional Neural Network (CNN) [161], like genetic algorithms, are therefore inspired by this natural phenomenon. Thus, CNN is a deep network composed of multiple layers and organized in blocks to leverage spatial information, it is therefore suited for classifying images. CNNs are composed of two major parts: the first part called the convolutional or feature learning and the second part called the classification part:

i. Convolutional or Feature Learning part: the network performs a series of convolutions and pooling operations in order to detect features. If we take the example of a picture of a cat, this is the part where the network would recognize its form and four paws;
ii. Classification part: this part is composed of fully connected layers which serve as a classifier of the extracted features. Otherwise, the matrix of the extracted features goes through a fully connected layer to classify the images.

Each input image pass through a series of convolution layers with filters (Kernels), Pooling, fully connected layers and apply Softmax function to classify an object. The figure 2.8 represent a complete CNN to process an input image and classifies the objects. As

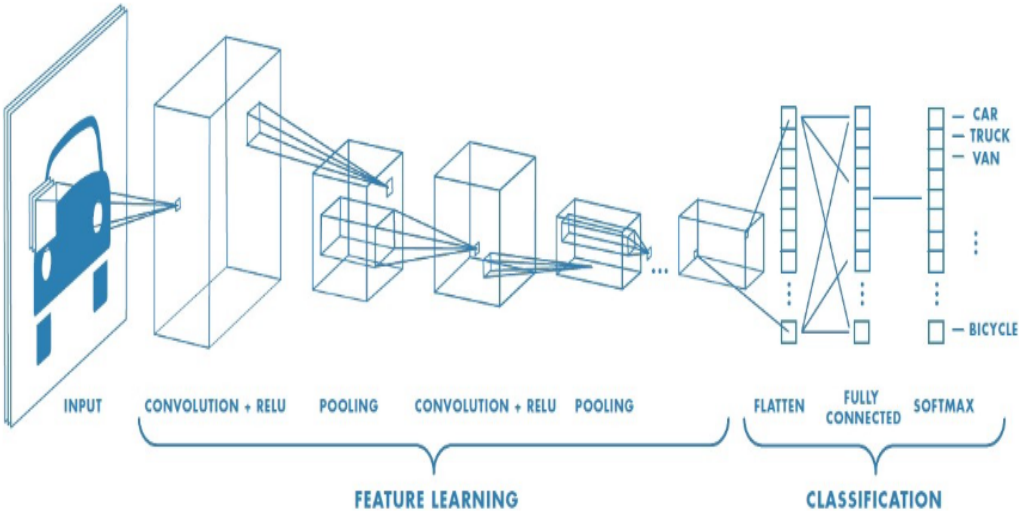

Figure 2.8: Representation of a CNN [231].

we described above, a simple CNN is a sequence of heterogenous layers, and every layer transforms one volume of activations to another through a differentiable function. Three main types of layers are stacked to build CNN architectures: Convolutional Layer, Pooling Layer, and Fully Connected Layer:

i. Input layer: holds the raw pixel values of the image;
ii. Convolution layer: it is the core building block of a CNN. It consists in preserving the relationship between pixels values by learning image features. It creates a feature map to predict the class probabilities for each feature by applying a filter that scans the whole image;
iii. Pooling layer: scales down the amount of information generated for each feature from the convolutional layer and maintains the most essential information (the process of the convolutional and pooling layers are usually repeated several times);
iv. Fully connected layer: uses the output of the convolution layer to determine the best class for the image based on the final probabilities.

2.2.2 Machine learning for time series

Time-series are simply a set of ordered data points with respect to time [99]. Different algorithms are used to extract certain statistical information to predict the future values based on stored past time-series data. We distinguish two different kinds of time series:

(a) Univariate time series: include datasets where only a single variable is observed at each time. For example, a dataset of the temperature values at each hour for the past 2 years. Here, temperature is the dependent variable on time which means predicting the temperature for the next few days consists in looking at the past values of the temperature;

(b) Multivariate time series: include datasets where two or more variables are observed at each time. For example, a dataset composed of the temperature values, wind speed, cloud cover percentage, etc. for the past two years. In this case, there are multiple variables to be considered to optimally predict temperature.

Before introducing the different types of ML algorithms for time series, we go through the formal definition of time series: A dataset $D = \{(X_1, Y_1), (X_2, Y_2), ..., (X_N, Y_N)\}$ is a collection of pairs (X_i, Y_i) where X_i could either be a univariate or multivariate time series and Y_i is its corresponding label. The goal of time series classification consists in training a classifier on a dataset D to map from the space of possible inputs to a probability distribution over the labels [99].
Time Series Classification is an important and challenging problem in data mining [99]. Classical ML methods are not suitable for time series classification.
CNN can discover and extract the suitable internal structure to generate deep features of the input time series automatically by using convolution and pooling operations [290]. In [296], CNN has shown its superior ability on the task of measuring patient similarity. Authors in [258] have built an accurate personalized prediction model with the learned similarities time-fusion CNN based framework to account for the temporality across different time intervals. However, one drawback of the traditional CNN architecture is that it can not fully utilize the temporal and contextual information of EHRs for disease prediction.Consequently, modeling temporality and content of EHR data is more challenging with a CNN.
Models such as kNN, MLP etc. have been shown to be effective in various automatic classification applications. Indeed, these models (called "Shallow ML algorithms" in this manuscript) learn to assign a label to new data by exploiting their characteristics which are expressed in the form of a set of values. Despite their interpretability, DTs are not suitable for processing temporal data and they are particularly intolerant to missing data problems which is a common challenge in sequential data. Theoretically, the traditional SVM is not able to handle with temporal dependencies since it considers data as independent and identically distributed.

Therefore, classical methods generally are not adapted to time series data. They consider time series as a vector of characteristics and often examine each event independently of the others without capturing the relationship between events. In this case, modeling time series can be considered in an external way (outside of the model) by using for example sliding time window technique or recurrent sliding windows technique.
Another more efficient way to solve sequential learning problems is to model time series inside the algorithm. One of the most efficient method is the recurrent Neural Networks (RNNs).

The Sliding Window Method

Sliding Window method is considered as a temporary approximation of the time series data. It consists in converting the sequential learning problem into a classical learning problem by constructing a window classifier able to map an input window to an output value. In fact, time series dataset is restructured as supervised learning problems by using the value at the previous time step to predict the value at the next time-step. The idea is to slide the time window of fixed size. Figure 2.9 shows an example of the process of sliding window with window size=4. Given a multivariate time series X_i, with $1 \leq i \leq 10$, its label Y_i. The dataset $D = \{(X_1, Y_1), (X_2, Y_2)..., (X_{10}, Y_{10})\}$ describes 10 visits of a given patient from time 1 to 10. The idea is to construct a window classifier w to predict the output Y_i at t by using the window. The window is moved by one position to the right and it shows another subsequence to which the processing can be applied. Initially window has covered 5 historical data of 5 visits. Visits from X_1 to X_5 are used to predict the label of the next visit $Y6$. Then, window slides right side by one visit to cover other 5 visits (from X_2 to X_6) to predict the label of next visit Y_7. The process continues till time series data of a particular period considered is exhausted. The Sliding Window technique can be used to predict values in time series, using a classical ML algorithm. Classical ML methods need to fix the number of previous observed values as inputs of the model for each training process while the output is the forecasted values of the time series.
Sliding window technique is a good way to make any classical supervised algorithm able to solve sequential learning problem. However, this method does not consider the relationship between the predicted Y_t. It considers the relationship between the time points X_t without capturing those among the Y_t.
To overcome the weakness of sliding window methods and improve them, recurrent sliding technique can be used. Unlike the sliding window method, in recurrent sliding window method, the

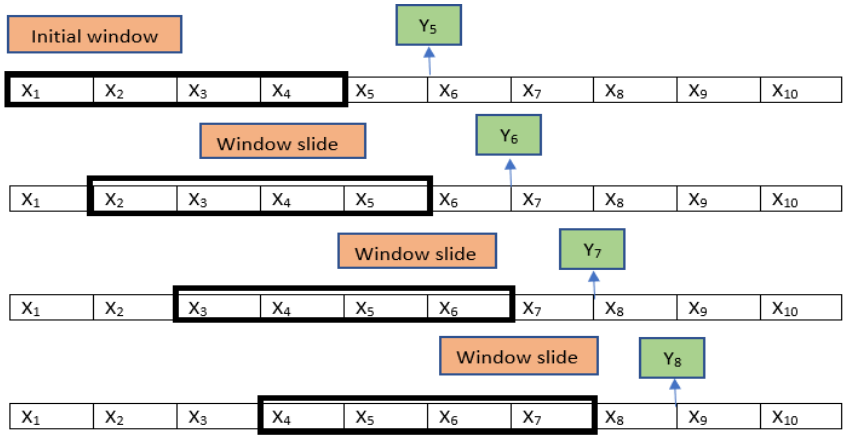

Figure 2.9: Process of sliding window.

predicted value Y_i is fed as an input to help making the prediction for Y_{i+1}.
In the literature, there are many approaches that integrate the temporal information inside the ML algorithm. In this work, we focus on Recurrent neural networks.

Reccurrent neural network

Recurrent Neural Networks (RNN) [139] have been used for sequence processing, particularly for predictive tasks. RNNs have a memory of what has been calculated in the past and this makes them particularly suitable for processing sequences. In theory, RNNs can retain information seen in a large sequence, but in practice, they lose their effectiveness over very long-term dynamics. It is in this sense that recent works have seen the emergence of recurrent neural network architectures with "gate" mechanisms that can significantly improve the memorization capacities of the models. More specifically, certain types of recurrent networks, Long Short-Term Memory (LSTM) [130] and Gated Recurrent Units (GRU) [22] have been shown to be particularly effective in modeling sequences whose dynamics could extend far over time.
Unlike MLP, RNNs shown in Figure 2.10, have cycles in their connectivity graph [94]. The main motivation behind this type of architecture is to be able to manipulate sequences of input vectors, each representing a temporal event, and not just isolated data having no temporal significance. By rolling out, with respect to time, the compact modeling of a RNN (see Figure 2.10), this type of network can thus be considered as a time series of MLP networks linked together through their respective hidden layers. This link allows RNNs to encode latent dependencies between events in a sequence of input vectors. These models were often acclaimed in particular for time series

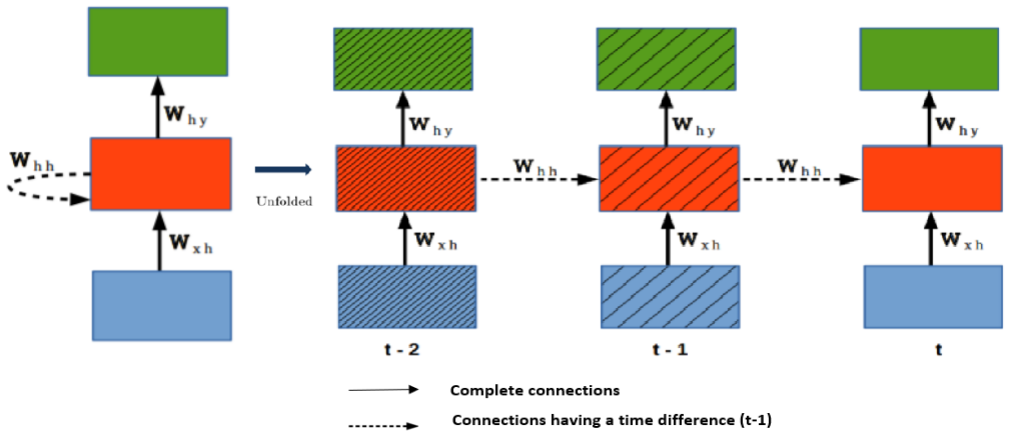

Figure 2.10: Unfolded representation and Compact representation of RNN.

and sequences, because their characteristics allow them to learn, store and take into account the contextual information passed during the processing of information. According to this modeling,

an RNN takes as input a sequence of events $x = (x_1, x_2, ..., x_T)$ and defines the sequence of hidden states $h = (h_1, h_2, ..., h_T)$ to produce the sequence of output vectors $y = (y_1, y_2, ..., y_T)$ by iterating from $t = 1$ to T:

$$h_t = f_h(W_{xh}x_t + W_{hh}h_{t-1} + b_h) \tag{2.19}$$

$$y_t = W_{hy}h_t + b_y \tag{2.20}$$

where T is the total number of input vectors, W_{xh}, W_{hh} and W_{hy} are the weight matrices, f_h is the hidden unit activation function which is generally the hyperbolic tangent and b_y, b_h are the biaises.
Being designed for non-looped networks, the gradient backpropagation algorithm is not sufficient for considering the time links expressed through the two previous formulas. One solution to this problem is to consider an unfolded "hierarchical" representation of the RNN. In the diagram illustrated in Figure 2.10, the time scale, represented through the diagonal arcs, carries a hierarchical meaning in the sense that the target layers are of higher level. It is through this hierarchical representation that the algorithm of backpropagation of the gradient generalized to non-looped neural networks is used. This version is called BackproPagation Through Time (BPTT) [122]. The particularity of this representation compared to a "classic" non-looped neural network is the existence of several shared parameters. For example, a common weight matrix W_{hh} passes information through the diagonal arcs. In addition, the weight matrices W_{xh} and W_{hy}, represented by the vertical arcs (Figure 2.10), are shared over time respectively. Based on the above, the learning algorithm based on the BPTT includes 3 steps:

(a) Forward propagation: information flows as in a normal non-looped network, from bottom to top. At each instant t (varying from 1 to T), the value of the hidden state at the previous instant (h_{t-1}) as well as the input vector of the instant t (x_t) are used to determine the new hidden state h_t. From this, the output vector y_t is calculated;

(b) Backwards propagation: once the forward propagation is completed, the error between the output y_t given by the network and the desired output for this sample is calculated. The error is then backwards propagated.

(c) Adaptation of parameters: the weights are updated in all the layers. For each neuron j in a hidden or output layer l which receives connections from neurons i in layer k, the bias and the weights are updated.

The advantage of RNNs lies in their ability to consider the past context when processing current information. However, these networks have difficulties in processing relatively long sequences, in particular those containing more than 10 events [131]. The gradient computation involves recurrent multiplication of the weights. Indeed, with cumulative calculations over the long term, the error obtained with the backpropagation of the gradient decreases or, less frequently, increases exponentially. The value of these derivatives may be so small, effectively preventing the weight from changing its value. In the worst case, this may completely stop the neural network from further training. This may cause the vanishing gradient or exploding gradients [131]. The dissipation or explosion of the gradient gets worse in this case depending on the number of layers.

To solve this problem, one solution is to replace the classic recurring unit with a recurring unit using gates. These gates are activating functions modulating the information flow in the unit. There are two types: GRU [65] and (LSTM) [130].

(a) Gated Recurrent Unit

GRUs [65] are improved version of standard RNN. GRUs are adapted to solve the vanishing gradient problem that can come with standard RNNs by using an update gate and a reset gate. The update gate controls information that flows into memory, and the reset gate controls the information that flows out of memory. The update gate and reset gate

are two vectors that decide which information will get passed on to the output. They can be trained to keep information from the past or remove information that is irrelevant to the prediction. As we can see in the figure 2.11, two inputs h_{t-1} and x_t are fed to the

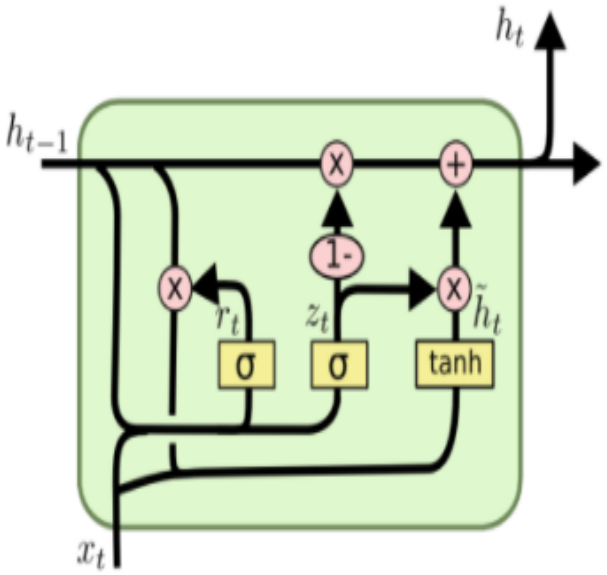

Figure 2.11: Gated Recurrent unit (GRU) [13].

unit. The input x_t is combined with the information coming from h_{t-1}, the memory of the previous state. The reset gate r, decides to keep the relevant information and forget the non relevent piece of information. The update gate z and the reset gate r_t are computed as follows:

$$z_t = \sigma(x_t W_{xz} + h_{t-1} W_{hz} + b_z) \tag{2.21}$$

$$r_t = \sigma(x_t W_{xr} + h_{t-1} W_{hr} + b_r) \tag{2.22}$$

A new memory content called current memory content is introduced. It uses the reset gate to store the relevant information from the past. It is calculated as follows:

$$\tilde{h_t} = tanh(x_t W_{xt} + h_{t-1} W_{hh} + b_h) \tag{2.23}$$

At each time step, a vector h_t that holds information for the current unit and passes it down to the network needs to be calculated as follows:

$$h_t = tanh(x_t W_{xt} + (r_t \odot h_{t-1}) W_{hh} + b_h) \tag{2.24}$$

Where x_t is the input at time t, W_{xz}, W_{xr}, W_{hz}, W_{hr}, W_{xt} and W_{hh} are the weights. b_z, b_r and b_h are the biases. $\odot$ is an element-wise product between the reset gate r_t and h_{t-1} which determines what to remove from the previous time steps.
Due to the update and the reset gates, GRUs can store and filter the information. So, the vanishing gradient problem is eliminated since the model is not washing out the new input every single time but keeps the relevant information and passes it down to the next time steps of the network.

(b) Long Short Time Memory

LSTM [130] is considered as a variant of the RNN. LSTM is one of the earliest approaches that address the challenge of short-term input skipping in latent variable models and long-term information preservation. Its architecture is like GRU. However, LSTM has one more gate that makes it more complex than GRU. Compared to GRU, LSTM is more powerful when there is enough data. However, GRU can train fewer data and it is faster than LSTM because it has less gates. An LSTM memory block consists of a memory cell c and three

multiplicative gates which regulate the state of the cell, forget gate f, input gate i and output gate o. The memory cell encodes the knowledge of the inputs that have been observed up to that time step. The forget gate controls whether the old information should be retained or forgotten. The input gate regulates whether new information should be added to the cell state while the output gate controls which parts of the new cell state to output. The gates are computed as follows:

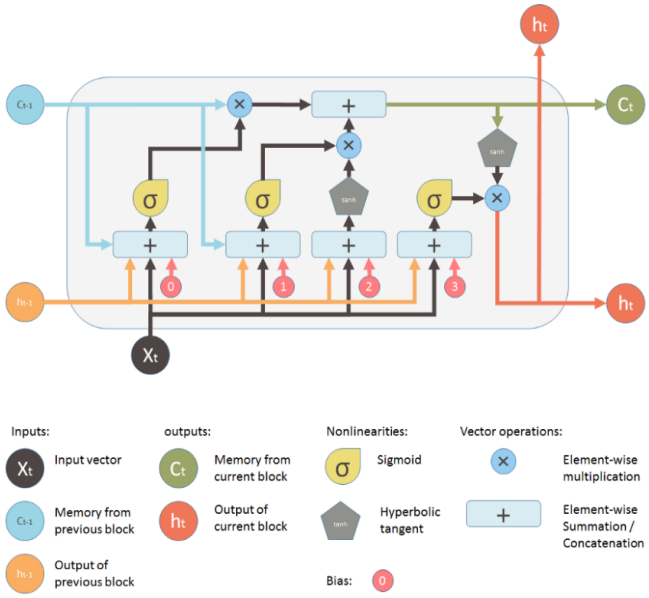

Figure 2.12: A Long Short-Term Memory (LSTM) cell [282].

$$f_t = \sigma(W_{fx}x_t + W_{fh}h_{t-1} + b_f) \tag{2.25}$$

$$i_t = \sigma(W_{ix}x_t + W_{ih}h_{t-1} + b_i) \tag{2.26}$$

$$o_t = \sigma(W_{ox}x_t + W_{oh}h_{t-1} + b_o) \tag{2.27}$$

$$c'_t = tanh(W_{cx}x_t + W_{ch}h_{t-1} + b_c) \tag{2.28}$$

$$c_t = f_t.c_{t-1} + i_t.c'_t \tag{2.29}$$

$$h_t = o_t.tanh(c_t) \tag{2.30}$$

The network takes three inputs: x_t, h_{t-1} and c_{t-1}. x_t is the input of the current time step, h_{t-1} is the output of the previous LSTM unit and c_{t-1} is the "memory" of the previous unit, which is considered as the most important input. As for outputs, h_t is the output of the current network. c_t is the memory of the current unit. f_t, i_t and o_t refers respectively to the forget, input and output gate that drop or retain the relevant information from previous states. The new hidden state h_t depends only on the input x_t and h_{t-1}. The hidden state in LSTM is dependent on gate states and an intermediate memory unit c'_t. The LSTM unit updates its state h_t and b_t (the bias at time t) at every time step t and propagates it to following dense layers. Therefore, this single unit makes decision by considering the current input, the previous output, and the previous memory. It generates a new output and alters its memory. Finally, the output is generated for this LSTM unit. This step has an output valve that is controlled by the new memory, the previous output h_{t-1}, the input x_t and a bias vector. This output valve controls how much new memory should output to the next LSTM unit. The final output of the model is:

$$y_t = \sigma(W_t h + b_t) \tag{2.31}$$

LSTMs have shown their effectiveness in various fields of application. They are currently considered as the state of the art approach in several tasks dealing with sequential and

temporal data [232] [103]. Their contribution is most evident in the case of fairly long sequences of events [171] [47].

2.2.3 Interpretation of Deep learning algorithms

As said before, deep neural networks have experienced strong predictive performance in recent years in many areas such as image recognition, textual and voice analysis. However, these good predictive results are generally accompanied by a difficulty in interpreting the model generation process on the one hand, and the learned result on the other. This "black box" effect of neural networks poses constraints in their use for example when they are used in critical domains such as healthcare [288] [284]. One of the first things that must be learned in ML is that there is often a balance between a model's performance and its interpretability. Transparent models (interpretable by a human, such as linear regression or decision trees) are generally less efficient than black box models, i.e. which are not directly interpretable by the human. The interpretability generally refers to the ability to explain or present information in humanly understandable terms. Different dualisms allow to map the main methods of interpretation developed in the discipline of DL.

Interpretation by extracting approximators

A global approximator is an interpretable model trained to imitate the prediction results of the black box model. Generally, sparse linear regression models, or decision trees and therefore decision rules are used as approximators. It is then assumed that the indicators used to interpret the approximators are representative of the complex mechanisms of the "black box" model. In the studied literature, there are many interpretable models that can be extracted from DL such as decision tree [73], or rules [142]. Different rule types can be extracted from neural networks we can cite:

- Logic rules, called also, "if-then" rules are the typical rules used by expert systems. The "if" part is the combination of conditions on the input variables, also named premises, and the "then" part is the conclusion;
- "M-of-N" rules were given a set of n conditions, if m of them are verified, then the consequence of the rule is considered true.

In recent years, many approaches for rule extraction from trained neural networks have been developed. According to Andrews et al. [14], the techniques of the rule extraction can be grouped into three main approaches namely decompositional, pedagogical and eclectic. The decompositional approaches [236], [107], [297] extract the symbolic rules by analyzing the activation and weights of the hidden layers of the neural network. The pedagogical approaches [74], [19] extract rules that represent input-output relationship so as to reproduce how the neural networks learned the relationship. The eclectic approaches [37], [133], [166] are a combination of the decompositional and the pedagogical approaches.
Tran and Garcez [265] proposed the first rule extraction algorithm from Deep Belief Networks. However, these stochastic networks behave very differently from the MLP, which are deterministic. Zilke et al. [297] have proposed an algorithm that uses a decompositional approach for extracting rules from Deep Neural Networks (DNNs) called DeepRED. This algorithm is an extension of the CRED algorithm [169]. For each class, it extracts rules by going through the hidden layers in a descending order. Then, it merges all the rules of a class in order to obtain the set of rules that describe the output layer based on the inputs. Bologna et al. [37] proposed a Discretized Interpretable Multilayer Perceptron (DIMLP) that uses an eclectic approach to represent MLP architectures. It estimates discriminant hyperplanes using decision trees. The rules are defined by the paths between the root and the leaves of the resulting decision trees. A pruning strategy was proposed to reduce the sets of rules and premises.
Craven et al. [72] proposed a method to explain the behavior of a neural network by transforming rule extraction into a learning problem. In other words, it consists in testing if an input from

the original training data with its outcome is not covered by the set of rules, then a conjunctive (or M-of-N) rule is formed from considering all the possible antecedents. The procedure ends when all the target classes have been processed. Zhou et al. [293] introduce an approach named REFNE that extracts symbolic rules from trained neural network ensembles that perform classification task. REFNE uses the trained ensembles to generate instances and then extracts symbolic rules from those instances. It avoids useless discretization of continuous attributes, by applying a particular discretization leading to discretize different continuous attributes using different intervals.
The rules extracted from neural networks must imitate the behavior of the model. According to [41], to measure the quality of the extracted rules, four main criteria can be considered:

- Accuracy: determined by the number of correctly classified test samples by a given rule;
- Fidelity: indicates the degree of matching between network classifications and rules' classifications. Practically, it is the fraction of instances on which neural network and the extracted rules give the same output. The fidelity must be higher to be sure that the extracted rules imitate well the behavior of the neural networks;
- Comprehensibility: is measured by the size of the rule set and by the number of antecedents. A rule that contains a lot of antecedents may be understandable;
- Complexity: indicates the difficulty of the implementation of the algorithm.

Interpretation by local approximators

Instead of looking at the model and trying to come up with global explanations, there are also, methods that look at every single prediction and then try to explain it.
Local Interpretable Model-Agnostic Explanations (LIME) [224] is one of these methods. This algorithm creates a model around a given prediction to approximate it locally. More precisely, LIME generates new data, namely data close to the prediction to be explained, then learns them using an interpretable model (such as linear regression or decision tree) and the classification made by any "black box" model. The objective of LIME is to gain the trust of users for individual predictions and then to trust the model. The disadvantage of the LIME method is that it does not provide a theory for generalizing the interpretability from the local model at a more global level.
Other techniques, such as the Shapley method [247], allow explaining a local decision while proposing, unlike LIME, an axiomatic theory (models that can be used for any class of learning method) on which to base the interpretability. The Shapley method proposes a classification of the contributions of input features according to principles derived from game theory.
Shapley method is very costly in terms of calculation, a SHAP variant [167] has been proposed on the same basis.
The saliency maps are methods specific to the imagery or the analysis of texts allowing to visually highlight (using mask) the parts of images or of the text which significantly participated in the decision of the "black box" algorithm. The calculation of the salience map is based on the learning algorithm (representation of gradients), the method is not agnostic to families of "black box" algorithms.
Another technique for interpreting DNNs by explaining their predictions is called Layer-wise Relevance Propagation (LRP) [20]. LRP is mostly applied to various data types (such as images, text, audio, video, signals) and neural architectures like CNNs and LSTMs. It operates by propagating the prediction backward in the neural network, using a set of purposely designed propagation rules [189].

Interpretation based attention

Inspired by human visual attention, an attention mechanism is the ability to learn to focus on specific parts of complex data, for example a part of a picture or a word in a sentence. Attention

mechanisms can be incorporated into natural language processing and image recognition architectures to help an artificial neural network learning what to "focus on" when making predictions. Attention in DL can be broadly interpreted as a vector of importance weights to predict or infer one element such as a feature in a patients' visit. Two kinds of attention mechanism are distinguished in LSTM:

(a) Global attention
All the hidden states considered when deriving the context vector c_t. As shown in figure 2.13 (A), the model infers a variable length alignment weight vector called a_t at each time step t based on the current state h_t and all source states $\overline{h_s}$. Thus, the global context vector c_t is computed as the weighted average, over all the source states, according to a_t [168];

(b) Local Attention
The local attentional mechanism chooses to focus only on a small subset of the source positions per target. As shown in figure 2.13 (B), the model first predicts a single aligned position p_t for the current state. To compute the context vector c_t, a window centered around the source position p_t is then used. The weights a_t are inferred from the current state h_t and those source states $\overline{h_s}$ in the window [168].

The diffrence between global and local attention mechanism is that Local attention attend to only a subset of targets while global attention attends to all the input targets.

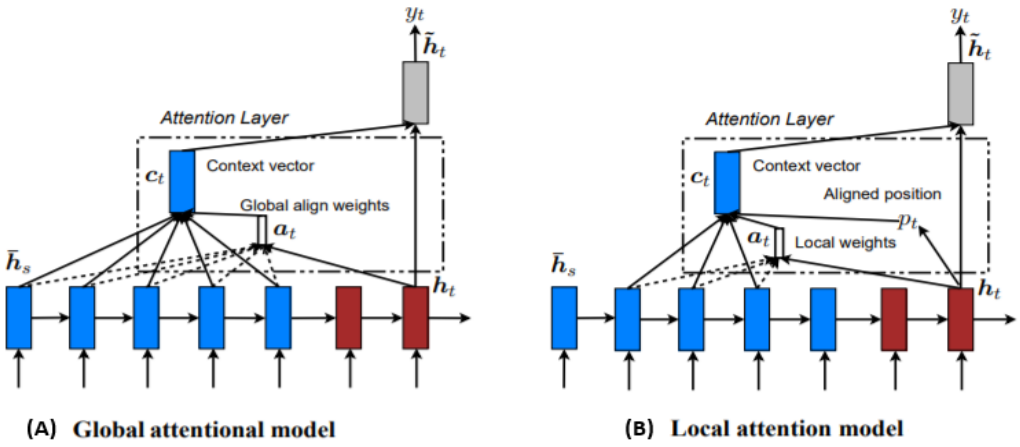

Figure 2.13: Attention-based Models: Global vs local attention [168].

This mechanism aims to resolve issues such as noisy variables in the multivariate time series. Specifically, the attention weights select those variables that are helpful for predictions. This aims to select the useful information across the various feature of the time series data for predicting the target.

2.3 Multiagent systems and argumentation

In the previous part we presented the necessary background related to ML. This part focuses on MASs and argumentation. We first present the definition of an agent in a MAS and the way in which the properties of intelligence and autonomy can be expressed in this context. Then, we present the concept of interaction, a central element for MASs since it represents what makes it possible to build a collective response from individual responses. The key concepts of Argumentation will be described, before focusing in multiagent argumentation.

2.3.1 Multiagent systems

Before defining what is a MAS is, it is important to first focus on what agents are and their characteristics.

(a) Agent
There is no formal and consensual definition of an agent, rather several complementary definitions have grown during the 90s ([181], [129], [251], [100], [105]). The most famous one is the definition of M. Wooldridge [278]. According to this author, "An agent is a computer system that is situated in some environment, and that is capable of autonomous action in this environment in order to meet its design objectives". More recently, Russell and Norvig [227] proposed a general definition which states that "an agent is anything that can be considered able to perceive its environment through sensors and act on this environment through actuators". This definition supposes that other agents are considered as part of the global environment. The whole definitions agree that an agent exhibits the following characteristics:

- Autonomy: an agent operates without any central intervention, and have some of control over its actions and internal state;
- Social ability: agents interact with other agents (and possibly humans or robots) via message exchanging and/or shared environment. For that purpose, a communication language is needed;
- Reactivity: agents perceive their environment (which may be the physical world, a user via a graphical user interface, a collection of other agents, the Internet, or may be all of these), and respond to changes that occur in it;
- Pro-activeness: agents do not simply act by responding to their environment, they are able to exhibit goal-directed behavior by taking the initiative. These goals should have been defined by an external set-up mechanism or inferred by the agents themselves.

(b) Multiagent system
A MAS is defined as a system made up of several agents which interact with each other in a common environment [279] [278]. Some of these agents can be people or their representatives (avatars), or even mechanical machines. If there are less than three agents, we are talking more about human/machine or machine/machine interaction than about MASs. In MAS, the communication dimension is important. Indeed, according to Genesereth and Ketchpel [112] an entity is a software agent if and only if it communicates correctly in an agent communication language. The architecture of a MAS is illustrated in Figure 2.14.

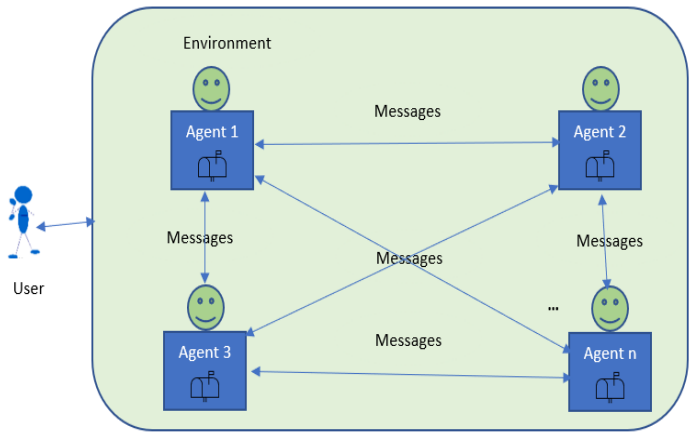

Figure 2.14: General architecture of a MAS.

Agents are generally classified into two main families: cognitive and reactive agents. However, it is common to design hybrid agents by combining cognitive and reactive capacities.

(a) A reactive agent: operates based on a simple correspondence between situations and actions (interacts with its environment but without reasoning about it). Its behavior is then governed by its relationship to those around it, without any internal state or representation of other agents or their environment. However, reactive agents can solve complex problems

by focusing on the modeling of an agent behaviour rather than the modeling of the agent [127]. The anthill, the termite, the beehive, and others are examples of systems based on reactive agents in the biological field. A reactive agent is often driven by motivational mechanisms (satisfaction of an internal need, achievement of a goal, etc.). It is also possible that a reactive agent responds only to stimuli from its environment. The general behavior of a reactive agent is described by a closed loop called perception-action (see Figure 2.15). Initially, the agent is in a certain internal configuration. Thanks to its sensors, the agent perceives a part of the environment in which it is immersed. Then, it chooses an action to be taken according to its internal configuration and its perceptions. This choice will be considered as the result of a decision-making function.

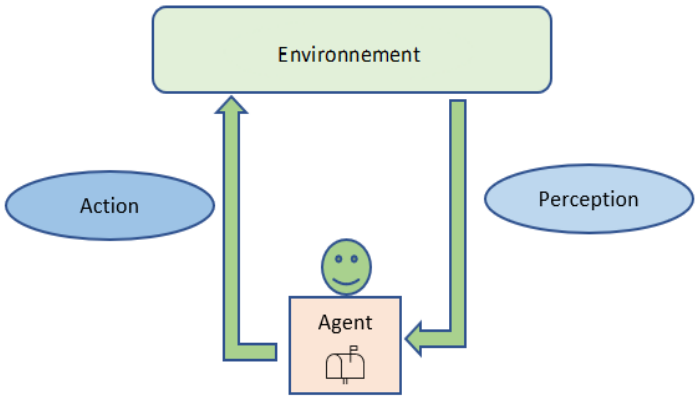

Figure 2.15: Perception-action loop of a reactive agent.

(b) A cognitive agent: is an agent with developed reasoning skills. Unlike the reactive agent, cognitive agent is generally intentional since it is characterized by the explicit representation of its objectives, an evolved representation of the environment and an ability to manipulate these representations to anticipate or reassess these objectives. BDI architectures are the most representative example of architecture for building a cognitive agent. The BDI agents [43] decide what actions to take from its internal states which are expressed in the form of beliefs (Belief), desires (Desire) and intentions (Intention). A cognitive agent is centered on three main functions: perception, decision-making, planning and action (see Figure 2.16).

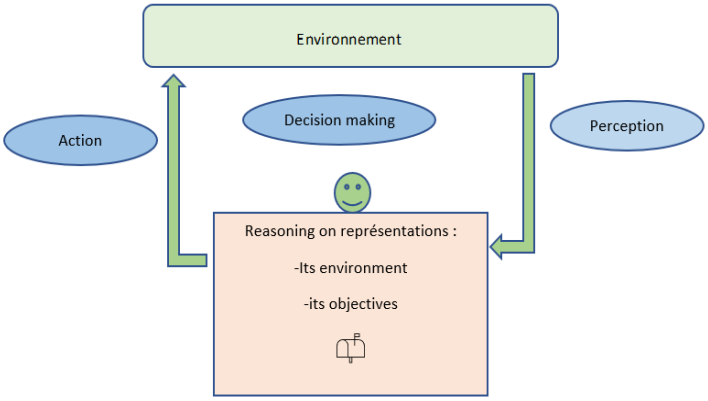

Figure 2.16: Behaviour of a cognitive agent.

(c) An hybrid agent: is a combination of cognitive and reactive capacities.

2.3.2 Agents interaction

The main characteristic of a MAS lies in the notion of interaction which can be defined as a dynamic connection of two or more agents through a set of reciprocal actions [101]. Interactions are thus expressed through a series of actions, the consequences of which in turn influence the future behavior of agents. The agents interact along a series of events during which the agents are in contact with each other, in a certain way, whether this contact is direct or takes place through another agent or the environment. The agents of the same system influence each other to construct a solution to the problem posed.
Interactions can be hard-coded by programmers according to the needs of the application or follow predefined protocols. With the growing number of dialogue protocols that have been suggested by various researchers comes the need to understand the properties of such protocols. Without this knowledge we have no basis for choosing between them, or even assessing whether they are adequate for a given purpose. We will present here the main:

- Coordination protocols: aim at serving one's own interests while trying to meet the overall goal of the system. In [141], the author characterizes coordination by two closely related aspects, namely commitments and conventions. Commitments provide the necessary structure for predictable interactions. Agreements control commitments in changing circumstances. As an example of a protocol for coordination we can cite the contractual network (contract-net) [255]. The advantage of such a protocol is that it achieves the coordination of tasks among the agents while ensuring the most optimal allocation;

- Cooperation protocols: consist in breaking down a problem into tasks and then distributing them [91]. Such an approach has the advantage of reducing the complexity of a problem. But there is a risk of having interactions between tasks and therefore of conflicts between agents. There are several mechanisms for distributing tasks. We cite the election mechanisms where tasks are assigned to agents following an agreement or a vote, contractual networks where tasks are assigned to agents following a cycles of calls for tenders or proposals. The multiagent planning assigns to planning agent's responsibility for the distribution of tasks and the organizational structure where agents have responsibilities for particular tasks;

- Negotiation protocols: are used when agents have different goals. The main negotiation mechanisms are: the language used, the protocol followed in the negotiation principle and the decision procedure that each agent uses to determine its positions, its concessions and its criteria for the agreement.

In the past years, several works have been proposed in agent interaction [82]. Most of them involve two parties' dialogues and the protocols they provide are based on two party dialogues. Even though, many specifications permit to send a message to more than one receiver, for example the FIPA ACL [194]. Multi-party dialogues [214] becomes more and more needed. It aims to ensure the interaction between several agents to find a solution for a problem. The idea of multi-party dialogue came from the fact that each agent might have a part of the solution, and their interaction might combine all the solutions [83].
Agents communicate with each other through a sequence of related rules on the same topic and not just with one-shot messages. For exchanging messages, they require a language with:

- A syntax: a common language to represent information and requests;

- A semantics: a structured vocabulary and a shared framework of knowledge;

- Speech act: is something expressed by an agent that not only presents knowledge, but performs an action as well [242];

- Performatives: are a type of illocutionary act (Inform, Ask, Answer, Promise, Affirm, ...);

- A pragmatics specifying: specify who to communicate with and how to find it, how to initiate and maintain an exchange and the effect of the communication on the recipient.

2.3.3 Argumentation

After describing the different protocols for agents interaction, we are particularly interested in the argumentation as a protocol of negotiation since it matches our problem. Argumentation [268] can be defined as the interaction of different arguments for and against some conclusion. Over the last years, argumentation has been gaining increasing importance in MASs, mainly as a vehicle for facilitating "rational interaction" (i.e., interaction which involves the giving and receiving of reasons) [217]. This is because argumentation provides tools for designing, implementing, and analyzing sophisticated forms of interaction among rational agents. Argumentation can be seen as the interaction of different, potentially conflicting knowledge or information considered as arguments, for the sake of arriving at a consistent conclusion. The fact that reasoning can be challenged has created an argumentative process. An argumentative process consists in developing a process of confrontation of points of view and deliberation which can lead to a decision.
We will detail below the different stages formalizing the argumentative process according to the description provided in [213].

Argumentative process

Overall, argumentation is the process of extracting a rationally justifiable position from incompatible starting points. It can thus be a process of reasoning in various essential stages which can be repeated by adding new knowledge. Generally, an argumentative process can be considered as a sequence of steps that begins with a knowledge base containing conflicts. This dialectical process includes:

- The speaker: is an actor (a person or an artificial agent) who is arguing. Such an actor is called the speaker or the orator;
- An argument: is the opinion defended by the speaker and shaped to convince, persuade, or deliberate;
- The audience: is composed of the actors which the speaker wants to influence in order to convince them to adhere to his opinion. All the argumentation is organized around an audience [266].

The argumentative process consists of four steps: (a) Arguments building based on generating arguments from the knowledge base, (b) Conflicts definition which consists in identifying conflicts between generated arguments, (c) Evaluation of the acceptability of different arguments and finally (d) Terminating which aims to select the justified conclusions (see Figure 2.17). We will detail below the different steps:

(a) Arguments building: in this step, arguments are generated from a knowledge base that contains the facts as well as a set of reasons which brings domain knowledge of a particular field to an agent. Otherwise, this step consists in arguments selecting. This notion generally refers to the concepts of explanation, proof and justification. The purpose of an argument is thus to support a proposition, a point of view or an opinion by citing reasons in its favor in order to convince an adversary, either to change its opinion or its judgment, or to incite it to act. An argument can have various forms, namely a part of a speech or an informal text in natural language given in a dialogue or a formal proof given in a well formalized logical language. From the knowledge base, a set of arguments can be selected. Concretely, an argument is a proof (deductive, analog, inductive, abductive). We can distinguish three different forms of this evidence. An evidence can be an inference tree which is a set of premises are linked in order to deduce a conclusion [164],[271], [280]]or a sequence of inferences which considers an argument as a set of rules linked together [202], [209], [210] or a pair of premises and conclusion, here the premises are considered as proof for the conclusion [96], [53], [7], [30].

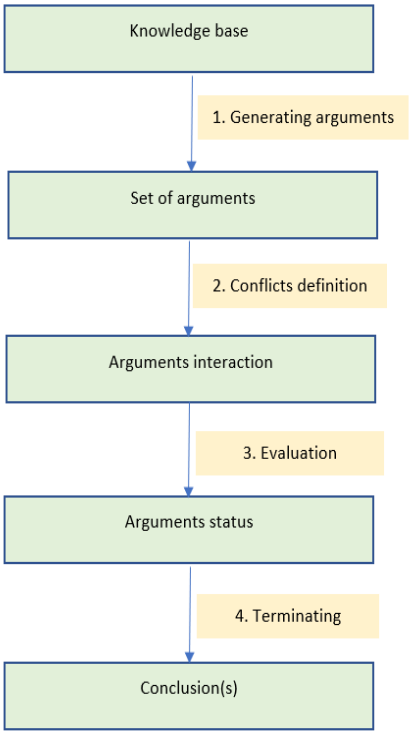

Figure 2.17: The argumentative process.

(b) Conflicts definition: this step consists in identifying conflicts between the generated set of arguments by ensuring arguments interaction. Argumentative interaction is defined as a situation of discursive confrontation in which antagonistic responses to a common question are constructed. Also, the question would be the contradiction of speeches, constitutive of the dispute, which gives rise to an argumentative question from which arguments and counter-arguments are organized. There can be several forms of interaction between the arguments [260]. Generally, the best known and most important of them are the Refutation and the Opposition.

- Refutation: two arguments which support contradictory opinions attack each other. Refutation consists in denying the conclusion of an argument by presenting a second argument with the opposite conclusion, for example "Titi is a bird then Titi flies" and "Titi is a penguin then Titi does not fly" are two arguments that refute each other. Indeed, the two arguments support two contradictory conclusions "Titi flies" and "Titi does not fly". This form of interaction is thus symmetrical. This attack relation is called disagreement in [253], reductio absurdum attack in [88], refutation by [6] and rebuttal in [31];
- Opposition: an argument supporting a thesis that contradicts one or more reasons that a second argument uses to support another thesis attacks the second argument. For example, any argument that supports the thesis "Titi is a penguin" attacks the argument "Titi is a bird and Titi is not a penguin so Titi flies". The opposition's relationship exists when the conclusion of one argument contradicts part of the premise of another argument. This relation is called differently in several argumentative systems. This notion of attack is called contradiction in [253], ground attack in [88], attack in [6] and undercut in [31]. Unlike the refutation, the opposition is not symmetrical because if a first argument opposes a second argument, the latter does not necessarily in turn oppose the first argument.

(c) Evaluation: after having identified all kinds of arguments interactions, it is necessary to assign an acceptability status to each argument present in the discussion. The status of an argument depends on its interactions with the involved arguments. By considering the interactions between the arguments, it is possible to assign a status to a conclusion supported by the argument. Several criteria can be considered to identify the status of

each argument included in an argumentative process. For example, in [31] the status are defined while considering the interactions between the arguments as well as the degree of conflict between the arguments. Whereas in [7] the criteria of weights and preferences between arguments are considered to assign a status to such an argument. Furthermore, in [134] the author assigns a probability to each argument which will be taken into account in judging the acceptability of an argument;

(d) Terminating: at the end of any argumentative process, among all the well-formed arguments, some will be distinguished, justified or even acceptable. This distinction can be made by taking into account the interactions between arguments and/or the evaluation of these arguments. Informally, the justified arguments are arguments that do not admit defeat or whose defeat is in turn defeated. Indeed, these justified arguments are those which could "win" in an argumentative dialogue between agents. Consequently, any opinion supported by an argument deemed acceptable could be considered as a conclusion or an exit from the argumentative process.

Abstract argumentation

Definition 1:
According to Dung [89] an abstract argumentation system is defined formally as a a tuple $S = \langle A, R \rangle$, where:

- A set of abstract arguments. Example: $A = \{a, b, c, d\}$;
- A binary relation on A, called attack relation. Example: $R = \{(a, b), (b, c), (d, c)\}$. aRb means that a attacks b.

A Dung-style argument system [89] can be represented by an oriented binary graph in which the nodes are arguments and the arcs correspond to the attack relation between arguments. Figure 2.18 shows an example of an abstract argument system represented as a direct binary graph.

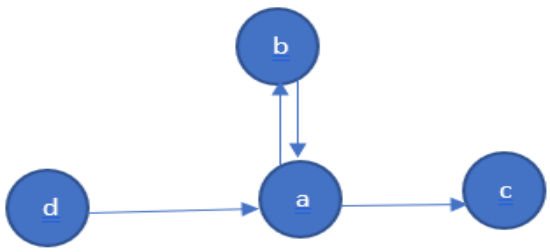

Figure 2.18: Binary graph of an abstract argument system.

Semantics of acceptance

The objective of an argumentation system is to evaluate the arguments interacting with each other and therefore provide a form of acceptability of arguments. Dung [89] have proposed different acceptability semantics to achieve this goal. Semantics identify subsets without conflict and which can defend themselves against any possible attacker. Three kinds of semantics exist for abstract argumentation. Extension based semantics, Labelling based semantics and Gradual valuation of arguments.

(a) Extentions based semantics:

Dung's goal was, starting from a set of arguments with an attack relationship, to define

Figure 2.19: Relation between semantics [214].

the acceptable set of arguments [89]. Such sets are called argument extensions. Different notions of extension have been defined by Dung leading to different policies for choosing the arguments to be accepted collectively. To do this, Dung offers different semantics to select the arguments deemed acceptable. We find conflict-free sets, allowable sets, naive extensions, stable extensions, preferred extensions, full extensions, basic extensions, etc. This notion of acceptability is said to be declarative or collective in the sense that this notion specifies how to decide that a set of arguments is acceptable. We begin by stating the notion of conflict-free set and the notion of acceptability of an argument [214].
Definition 2:
Let $S = \langle A, R \rangle$ an argumentation system where A is a set of abstract arguments, R is an attack relationship and $E \subseteq A$ is a subset of arguments from A.

- E is conflict-free if there is no attack between its arguments, formally : $\forall a, b \in E, (a, b) \notin R$;
- An argument $a \in A$ is acceptable with respect to $E \subseteq A$ if and only if E defends a, that is $\forall b \in A$ such that $(b, a) \in R, \exists c \in E$ such that $(c, b) \in R$.

Dung defined several semantics for building extensions. In the following we present some of the main used ones.
Definition 3:
Let $S = \langle A, R \rangle$ an argumentation system let $E \subseteq A$ be a conflict free set of arguments.

- E is an admissible extension if and only if it is conflict-free and all its arguments are acceptable with respect to E.
- E is a preferred extension if and only if it is a maximal admissible extension (with respect to $\subseteq$).
- E is a complete extension if and only if $\forall a \in A$, a is acceptable with respect to E ($a \in E$).
- E is a stable extension if and only if $\forall a \notin E$, $\exists y \in E$ such that yRa.
- C is a grounded extension if and only if it is the minimal complete extension (with respect to $\subseteq$).

Figure 2.19 shows inclusions between the extensions. For example, every stable extension is preferred; every preferred extension is complete; the grounded extension is complete... Many semantics have been proposed in [24]. There are two main ways to accept an argument. An argument can be credulously accepted or skeptically accepted. Otherwise, an

argument can be rejected.
Definition 4: Let $AS = \langle A, R \rangle$ an argumentation system and let $a \in A$. Argument a is:

- Credulously accepted: if and only if a belongs to at least one extension of AS under the S semantics;
- Skeptically accepted: if and only if a belongs to all the extensions of AS under the S semantics;
- Rejected: if and only if a does not belong to any extension of AS under the S semantics.

Refinements of these extensions have also been proposed such as recursive semantics [25], cautious semantics [69], semi-stable semantics [50], ideal semantics [90], CF2 semantics [108], equational semantics [108].

(b) Labelling based semantics:

Argument labelling allows to evaluate the status of arguments. Directly related to the extension-based approach, a labelling is a function that maps and associates every argument with a label. There are three values for labels [49]: in (the argument is accepted), out (the argument is rejected), or undec (the argument is undefined, not accepted or refused). Other approaches consider additional values [140].

(c) Gradual valuation of arguments:
Authors in [54] have presented the concept of gradual valuation in their framework where every abstract argument, receiving attacks, is assigned a gradual valuation. Initially, the arguments do not have an initial valuation. They are assigned gradual valuations when the attacks they receive are considered.
In a gradual valuation, let be an argument a which is assigned a gradual valuation $v(a)$. $v(a)$ is an element of a totally ordered set which has a minimum and a maximum element. In [54] two different types of gradual valuation have been proposed:

- Local valuation: the valuation of an argument depends on the valuations of the arguments which are set to attack it [214];
- Global valuation: the valuation of an argument depends on the set of attack branches leading to it [214].

With the development of the argumentation, several formalizations of argumentation theory have been proposed in the literature ([204], [58], [31]). These formalisms are in varying degrees of abstraction. They vary from naive representation of argument concept, where an argument is simply an abstract entity of which no information is provided on the internal logical structure, at a more complex degree of abstraction where knowledge and inference mechanisms are taken into account for the construction of arguments.
Dung ignores the first two stages of the argumentative process, namely generating arguments and identifying the nature of conflict between them. He is therefore not interested in the origin of the argument or its internal logical structure and he presents an argument as an abstract entity. Broadly speaking, argumentative approaches can be classified into four categories, starting from Dung's abstract argumentative systems [88] to hypothetical argumentation [39] and the unsuccessful argumentation of Garcia et al. [109] to arrive at deductive argumentative approaches like that of Pollock [202], Amgoud et al. [10] and the argumentative system of [30]. Many works have been developed in the literature which consist in extending these different classes of argumentation theory.
Many works have been proposed to refine and extend Dung's argumentative framework. Among which we can cite the preference-based argument system [[53], [8], [9]], the symmetric argument system [70], the bipolar argument system [55], the constraint argument system [71], hierarchical argumentation [186], the value-based argumentation system [27], Modgil's extended argument system [187], the abstract argument system with ASPIC structured arguments [208], the ASPIC + system [208], multi-sorted argument [225], probabilistic argumentation [134], abstract

argumentation theory based on quantified Boolean formulas [15] and the uniform argumentation system [197], etc. Generally, most of the developed works are based on abstract models and aim to define methods of selecting preferred arguments, as Dung.

Structured argumentation

Structured argumentation is an extension of Dung's abstract argumentation which gives a simple inner structure to arguments [261].
Several researchers have tried over the past decade to model the argument in a logical context. Thus, several formal models of argument have emerged in recent years. The main objective of these models is therefore to find a formal and concrete representation of the notion of argument from a logical language and a relation of logical deduction. In addition, different underlying logics have been used to define the concept of argument and identify relationships between arguments such as propositional logic [30], [10], first order logic [202], the logic of defects [234] and the logic of description [34], etc. Generally, these logical and structured argumentation systems are characterized by two crucial points. On the one hand, an argument is in the form of a couple made up of a set of logical formulas called premises and a formula deduced from these premises called conclusion. The premises can be considered as a proof for the conclusion Additional conditions are sometimes required, by some systems, in the construction of the arguments such as for example the coherence of the set of premises or the minimality of the set of premises for the inference of the conclusion, etc. On the other hand, interactions between arguments have a logically based definition. In particular, the conflict between arguments is essentially based on the notion of inconsistency in the considered logic.
In Structured argumentation, we suppose in general that an argument has an internal structure and include the following components:

- a set of hypotheses or premises;
- a method of reasoning or deduction;
- a conclusion.

Stephen Toulmin [264] defined the argument as the construction of an analytical scheme towards a conclusion using data and justifications. He distinguishes six basic bricks:

- The conclusion (claim): is the fact that needs to be justified;
- The data: are the facts on which the argument is based;
- The warrant which legitimizes the jump between the data and the conclusion;
- The backing, i.e. warranty support;
- The modal qualifier which allows to indicate the strength of the jump between the data and the conclusion;
- The exception condition (rebuttal) which appears when the jump between the data and the conclusion is not legitimate.

An argument is then said to be structured in the sense that normally the premises and claim of the argument are made explicit, and the relationship between the premises and claim is formally defined.

2.3.4 Multiagent argumentation

The advantage of using MAS is their ability to provide robustness and efficiency, to allow interoperability of existing systems and to solve problems for which data and expert knowledge are distributed. MAS are ideal systems for representing problems with multiple methods of solving, multiple perspectives, and/or multiple solvers. They also inherit the possible benefits of artificial intelligence such as symbolic processing (at the level of knowledge). To achieve their individual or collective goals, agents need to communicate. For that purpose, sophisticated interaction patterns are used. Negotiation protocols (see 2.3.2) can then be employed as interaction patterns for reaching an agreement acceptable to all involved parties.
Indeed, argumentation theory has been an inspiration for agent communication [174] since it allows exchanging arguments between agents, justifying their choices and providing reasons defending their claims. The added value of argumentation lies in the justification of positions [216] and the efficiency of agent's communication by letting them reveal relevant required information during a conversation.
We distinguish two different possible forms of using argumentation in MAS (multiagent argumentation). The first one consists in using argumentation by a single agent in order to reason about its beliefs and possible actions. The second one requires two or more agents. Each one provides its different beliefs and goals for exchanging with others to achieve the final goal.
The use of argumentation has been proven to be fruitful in negotiation dialogues [148], as well as in dialogues leading to a decision [147]. In our work, we focus on persuasion dialogues. In persuasion dialogues two or more participants try to resolve a conflict of opinion, each trying to persuade the other participants to adopt their point of view [207]. In these dialogues, every agent uses its beliefs to debate with other agents and try to convince them about the correctness of its point of view on the topic. Argumentation is often used in persuasion dialogues [205]. Parkken et al. [206] have provide a list of persuasion systems, with a brief analysis of each one's characteristics. The work proposed in [148] is an argumentation-based negotiation protocol where the negotiating parties and different arguments are linked. Different roles of agents and context of interaction are taken into consideration and the strength of the arguments are defined based on specific factors. During the negotiation, the agents can adapt their negotiation strategies.

2.4 Conclusion

This chapter has enabled us to talk about the different existing ML algorithms and their ability to consider the temporal aspect of EHRs in the medical field. These studies allowed us to choose the most adapted model for our research project. We also had the opportunity to understand the basic foundations of MAS and the argumentation process.
The goal of this project is to exploit the adaptation and reasoning distribution capacities of MAS and the efficiency of argumentation to introduce intelligence into the way of combining learning algorithms. In our approach we use neural networks which are known for their difficulty of interpretation. Multiagent argumentation will overcome this weakness by adding transparency and intelligence to the system. The idea is to extract knowledge from classifiers, then integrate them into agents for argumentation. This will provide a clear view of the system and provide an understandable decision for the user.
In the coming chapters, we will try to present some works in the literature using ML and multiagent argumentation for EHRs. We will focus on the works that add transparency to medical decision systems.

Chapter 3

State of the art

Contents

3.1 Introduction

The following two sections aim to highlight the richness of applications in the medical field involving the use of ML, MAS and argumentation. The first section focuses on the ML techniques used in healthcare. The second section will focus on the different types of existing works related to the MASs and argumentation in the medical field. Particularly interested in the transparency and the interpretability of support decision systems, we will focus on explainable systems.

3.2 Machine Learning in healthcare

In the care process, the diagnostic phase is essential for patient orientation and follow-up. ML brings new solutions to healthcare professionals to save time and optimize the correct diagnosis. It opens new perspectives in the detection of diseases. The goal is not to replace the doctor with the machine, but to support him/her in the analysis and interpretation of the huge volumes of data collected. ML also makes it possible to promote correct diagnoses and fight against medical errors by generating differential diagnoses and suggesting additional examinations. For example, it can help physicians more easily to predict disease [291], to extract medical concept [179], to group disease and patient [85], to model patient trajectory [92] and to support clinical decision [155], [183].
The main factor of the progress of ML made in medicine, is the facilitation of data collection and information waiting to be used. This huge amount of data is called EHRs [60]. EHRs have been used mainly to improve the efficiency and ease of access of health data [276]. But also, they have been used to setup intelligent ML systems. For example, Menachemi et al. have estimated that the introduction of EHRs can reduce serious processing errors by 55% and up to 83% when coupled with a decision support system [178].
However, the challenge in medical support systems is to use medical data to adapt treatment for each patient's profile. Recent medical analyzes produce many different data: clinical data, genomic data, metabolomic data, proteomic data or images... The challenge is to process data of large dimension and heterogeneous origin, it represents data integration.
For this purpose, different ML methods have been used in the literature. For example, classical ML techniques such as logistic regression [156], SVM [51] and random forest [215] have been employed for medical applications. However, these models do not deal with high-dimensional EHRs. The massive, irregular and complex nature of this data [219] requires a powerful modelling techniques that can discover and take into account complex nonlinear interactions among variables (i.e., a time series mixed-type and multimodal data packed in irregular intervals) [162], [117].

DL is a ML technique that provides high performance prediction results in various application areas.

The advantage of DL is its ability to analyze EHRs since it addresses the major challenges of EHR such as variable-size discrete inputs, irregular timing...). However, one of the recurring criticisms of DL algorithms is their lack of interpretability for users. Although it is known that information is abstracted by the different hidden layers, the problem lies in the interpretation of DL to explain predictions.
To classify works that address DL in healthcare, we can consider two different criteria: the kind of the used model and the interpretability.
As the list of works focusing on clinical applications resulting from recent advances in deep EHR learning remains wide, we will present a certain number of works that we have found interesting and reflecting the diversity of the problems addressed. This study of state of the art was carried out between 2015 and 2019. Table 3.1 summarizes the main selected approaches of DL for EHRs and illustrates their main characteristics in terms of the used model, considered features, task, interpretability and Attention mechanism (defined in section 2.2.3 of the chapter 2).

Approach	**Used model**	**Considered features**	**Task**	**interpretability**	**Attention mechanism**
Doctor AI	GRU	Diagnoses (ICD-9), procedures (CPT), medications (GPI)	Prediction of the medical codes in future visits and the duration until next visit	NO	NO
Med2vect	MLP	Diagnoses (ICD-9), procedures (CPT), medications (NDC)	Prediction of the medical codes in previous/future visits	YES	NO
Deepr	CNN	Diagnoses (ACS), procedures (ACHI)	Hospital Readmission Prediction	YES	NO
DeepCare	LSTM	Diagnoses (ICD-10), procedures (ACHI), medications (ATC)	Disease progression, unplanned readmission prediction	NO	NO
DeepPatient	SDA	Demographic variables, diagnoses (ICD-9), medications, procedures, lab tests, free-text clinical notes	Multi-outcome Prediction	NO	NO
RETAIN	MLP / RNN	Diagnoses (ICD-9), procedures (CPT), medications (GPI)	Heart failure prediction	YES	NO
Ensemble model	RNN / FFNN	Demographics, provider orders, diagnoses, procedures, medications, laboratory values, vital signs, flowsheet data, free-text medical notes	Inpatient mortality, 30-day unplanned readmission, long length of stay, diagnoses	YES	YES
Dipole	RNN	ICD-9 diagnosis, procedures, medications	Predicting patients' future health information	YES	YES
Patient2Vec	GRU	Sequence of multiple medical codes	Hospitalization prediction	YES	YES
Deepsofa	GRU	Admissions variables,Demographics variables	Illness severity assesment	YES	YES
GRAM	RNN / Ontology	Diagnoses (ICD-9), admissions variables	Heart failure prediction task	YES	YES

Table 3.1: Some deep aproaches properties

3.2.1 Used model

Many DL architectures (e.g., feed-forward neural networks (FFNN), convolutional neural networks (CNN), and recurrent neural networks (RNN)), have been employed for treating EHRs.

Fully connected feed-forward neural networks

Unlike RNNs where the connections between the nodes form cycles, Fully connected feed-forward neural networks (FC-FFNN) [12] refer to neural networks where the information moves in only one direction: from the input nodes, through the hidden nodes to the output nodes like MLP, autoencoder...

(a) Multilayer Perceptron
In the litterature, MLP has been used for healthcare.
Med2Vec [61] have used MLP model for learning concept and patient visit representations. The proposed model used medical codes and demographic information as input to predict the medications of the visits and the neighboring visits in a context window.
Authors in [32] have used a DMLP to predict future risk with a certain probability. The proposed model extracts the relevant features for the prediction by adding L1 and L2 regularization [119] to the objective function. To validate their approach, the authors use three public datasets for two-class problem: Wisconsin Breast Cancer, SaHeart and Pima Indians Diabetes. Thus, they have showed that the number of selected features reflects the result of MLP classifier.
In [12], authors utilized a MLP in order to predict the delivery type of pregnant women. They show good experimental results on small EHR dataset but did not compare their approach to other ML algorithms.

(b) Autoencoders
Some approaches used a non-supervised learning model (such as Autoencoders) to learn low-dimensional representation of EHR. EHRs are considered complex since they contain high dimensional, temporal, mixed-type and multimodal data packed in irregular timestamps, that's why, it is necessary to build high level representation in order to make their use more simple by others classifiers.
Miotto et al. [182] proposed to learn deep patient representations from the EHRs for training a feature extractor using a three-layer Stacked Denoising Autoencoder (SDA). They applied this novel representation on disease risk prediction and used Random forest as classifier. The evaluation was performed on a dataset containing approximately 4.2 million (deidentified) patients. The dataset was generated from the Mount Sinai Health data. The results showed that the deep representation leads to significantly better predictions.
Nezhad et al. [190] used stacked autoencoders to learn high level representation of patient in order to predict the risk factors for heart damage from a subgroup of African and American population that had high risk of cardiovascular disease.
In [229], authors compared four unsupervised DL architectures: Stacked sparse autoencoders, Adversarial autoencoders, variational autoencoders and deep belief network on small and large EHR datasets. They showed in their experiments that stacked sparse autoencoders are preferable for small datasets due to sparsity regularization while variational autoencoders outperform the other architectures for large datasets due to their ability to learn distributions representation.

Convolutional Neural Network

CNN is also used to predict future clinical events by modeling the EHR as a temporal matrix where the horizontal dimension represents time, and the vertical dimension corresponds to medical events. CNN was used in in the work [57] to predict congestive heart failure and chronic

obstructive pulmonary disease. The proposed architecture is composed of four-layer CNN. EHR matrices are fed into the first layer. Phenotypes are extracted from the first layer in the convolution layer. The convolution operator is applied on the time dimension. Significant phenotypes are retained due to max pooling layer which introduces sparsity on the detected phenotypes.
Nguyen et al. present Deepr [192], a DL system that learns extracting features from medical records in order to predict medical risk for the patient. Deepr is a multi-layered architecture based on CNNs. It treats the irregularities of EHRs. Deepr can identify regular clinical motifs of irregular data. First, medical record with multiple visits (diagnosis, procedure) are sequenced into phrases separated by spatial "words" that corresponds to time-gaps. Then words are embedded into continuous vectors. In the convolution layer, local word vectors are convoluted to detect local motifs. Finally, Deepr derives a global feature vector using the pooling layer and applies a linear classifier to predict a future disease risk.
Suo et al. [259] proposed a Time-fusion CNN to learn patient representations and measure similarity among patients based on their historical records for disease prediction. First, they used a fully connected layers to embed each visit representation into a vector space. This allows to reduce feature dimensions. Then, they segmented the patient embedding matrix into several subframes. A one-side convolution and pooling are applied for each subframe to capture the sequential relation across adjacent visits. To obtain a global vector representation, a weighted average of the obtained vector representation of each subframe is computed. After this step, a similarity is learned. The resulting similarity scores are then used for disease prediction.
Xiaozheng et al. [163] proposed a DL framework for diagnosis prediction for pediatric Chinese EHRs. In this approach, a CNN is combined to Natural Language Processing (NLP). Unlike previous works, authors extracted knowledge from text. The framework is composed of three parts: word segmentation, word embedding and model training. First, authors generated the vector representation from medical dictionary which has been collected from EHRs. Then, they adopted Word2vec [203] in word embedding to construct the word vector representation of EHRs. Finally, the diagnosis with EHR data were fed into a CNN model for training. They have tested their framework by using different models including CNN models, RNN models and CNN-RNN hybrid architecture. The authors demonstrated that one-layer CNN performs best among CNNs with more than one layer with an average of accuracy and F1-score up to 81%.

Recurrent Neural Networks

RNNs are among the most powerful models which can process a sequence of inputs and retain its state while processing the next state. Among the approaches that used RNN, we can cite Pham et al. [199] which employed LSTM to exploit its ability to model long-term dependencies in sequences. The particularity of the proposed approach, called Deepcare, lies in the possibility of temporal parameterization which allows to model the dynamics of each disease. DeepCare addressed several challenges: variable-size, discrete inputs, confounding interactions between disease progression and intervention, and irregular timing. The learning phase was carried out using data from an Australian hospital. Pham et al. were able to achieve 79% and 74% accuracy in predicting unplanned readmission for diabetic patients (within 12 months) and for patients with mental disorders (within three months), respectively.
The system proposed by Lipton et al. [165] uses EHRs from patients who have passed through intensive care units. This service presents a wide variety of different patient profiles. The LSTM is used to model sequential clinical data and to classify diagnoses. Authors validated LSTM on multilabel classification of diagnosis by considering different pathologies.
Meanwhile, Doctor AI [60] intended to predict the diagnoses of the next visit, the treatments corresponding to the predicted symptoms, and the date of the next visit. A sequence of pairs (event, time) is embedded into a GRU network. At each point in time, the hidden units' weights are taken as the patient representation at that point in time. From this representation, Doctor AI can model and predict future patient status. It can learn efficient patient representation from a large amount of patient records and predict future events of patients. The performance of Doctor AI increases in proportion to the number of visits observed for the same patient. The results showed a sensitivity of 79.5% demonstrating the ability of the model to satisfactorily predict future diseases.

Authors in [16] have employed LSTM which uses expert features and contextual embedding of clinical concepts for predicting 30-day unscheduled readmission risk at each visit. Generated visit representations from human and machine-derived features are fed sequentially into the LSTM for training. In this work, authors addressed the class imbalance problem by using Cost-sensitive classification which aims to encode the penalty of misclassifying samples from a particular (minority) class [274] by embedding a cost matrix in the loss function. The framework has shown good results when using real data from over 7500 hospitalized patients in Sweden.

Models combination

Rather than building one model and hoping this model is the best/most accurate predictor some authors choose to combine different ML algorithms to improve ML results. A CNN was combined with other DL models to better treat medical data. As an example, we can cite the work proposed by Lauritsen et al. [160]. In their work, authors present a scalable DL approach for early sepsis detection on the heterogeneous data set that includes hospitalizations both within and outside of the medical centers. The model is structured as a CNN, followed by a LSTM also known as a CNN-LSTM model [84]. First, the model projects the sparse inputs into dense 1000-dimensional vectors, reducing the dimensionality for the following convolutional layer. Short-term temporal developments for a patient are then captured in the model by a stack of CNNs. Finally, the model captures the key factors and interactions from long-term temporal data by feeding the output from the convolutional blocks into an LSTM layer that incrementally builds up a representation of the temporal inputs and continually predicts an output.
Similarly, Landi et al. [158] have proposed ConvAE which combines CNN and autoencoders to process heterogeneous EHRs and generate patient representations by transforming patient trajectories into low-dimensional vectors. The learning model is based on word embeddings. First, an autoencoder is used to derive vector-based patient representations from a huge heterogenous EHR. Then, using a CNN, ConvAE integrated the temporal aspects of patient data. Authors have shown that the generated patient representations by ConvAE lead to clinically meaningful insights.
Zhang et al. [289] have introduced an approach for patient subgrouping by using three different kinds of ML algorithms: Autoencoder, RNN and kNN. First, they divided the clinical data into two groups: non-time series data and time series data. An autoencoder was used to learn patients' representations from non-time series data, and a RNN based Autoencoder was used to extract representations from time series data and capture the time irregularity. Finally, a weighted k-means method was introduced to subgroup patients with the pairwise representations. The proposed approach has shown good results in patient subgrouping when using real medical datasets.

3.2.2 Deep interpretable models in healthcare

Healthcare, a critical application area, requires understanding the automatic learning algorithms used as a decision support tool. This plays a fundamental role in understanding and implementing human-machine collaboration and limiting resistance to digital change. Interpreting models can generate a new point of view or analysis with high added value for the decision maker. Thus, an intelligible explanation of the mechanism and the results of the neural network can be a diagnostic and analytical tool for the operator in charge of the decision.
Previous works employed different strategies to develop explainable deep models for medical prediction. We focus particularly on the attention mechanism and rule extraction from the network.

Interpretation based on attention

Attention in DL can be broadly interpreted as a vector of importance weights to infer elements that contribute the most to the prediction. Attention mechanism in DL was used in healthcare. Ma et al. [170] have proposed Dipole, a model based bidirectional RNNs which can predict

patients' future health information. Dipole exploits the ability of bidirectional recurrent neural networks which keep in memory the information of both the past and the future visits. To interpret the prediction results and to better represent information from all visits, Dipole introduced three attention mechanisms: location-based, general, and concatenation-based. These mechanisms calculate the attention weights for all the prior visits of each patient, capture the correlations between different visits and provide the importance of each visit for the prediction. Thus, Dipole can interpret the prediction results and the learned medical code representations.
One of the recent DL models applied to medical data that both harnesses the performance of RNNs and preserves interpretability is RETAIN (REverse Time AttentIoN) developed by Edward Choi and al. [63] to predict outpatient heart failure. RETAIN can interpret visit and variable importance by using two RNNs and a two-level neural attention model to generate two sets of attention weights and to process sequential data. It calculates the attention weights for a visit at a given time, using the medical information in the current visit and the hidden state of the RNN at the same time point, to predict the visit at the next time point. As an improvement of RETAIN, Kwon et al [157] produced an interactive visual interface named RetainVis, that offers insights into how individual medical codes contribute to making risk predictions. RetainVis improves interpretability as well as interactivity of RETAIN and helps users to explore realworld EMRs, gain insights, and generate new hypotheses.
A GRU with hierarchical attention called GRNN-HA [245] was proposed for mortality prediction. Like RETAIN, the proposed model can calculate two levels of attention: attention weights for medical codes and attention weights for patient visits. The first attention level is introduced into the GRU when embedding medical records to enable the model to pay more attention to efficient codes. Similarly, the second attention is introduced in the visit-level to focus more on the visits that contribute more to the prediction.
Shickel et al. [250] have proposed DeepSOFA for illness severity. DeepSOFA leverages temporal measurements by using a GRU and self-attention to highlight the most important visits when formulating the mortality prediction. At each time point, after embedding EHR, the model learns to distribute its internal "attention" by assigning weights to all preceding time points. The attention scores are then used to determine and visualize the severity of the time series patterns.
Zhang et al. [287] have proposed Patient2Vec, a framework that compresses the entire patient EHR into a complete vector representation and learns an interpretable and personalized deep representation of EHR data. Patient2Vec used GRU and self-attention to predict the future risk of hospitalization. First, the model learns vector representations of the medical codes containing words by using word2vec approach [203]. Then, a self-attention mechanism is employed for training the network to learn the weights. The learned weights are aggregated into one vector to provide a comprehensive representation. Finally, a logistic regression layer uses vector representation of a patients for the prediction of outcome. Patient2Vec can not only outperform the baseline methods and produce a vector space with meaningful structure but also visualize and interpret the learned feature importance.
Unlike previous works, Kaji et al. [146] applied an attention mechanism at the level of input variables, then used LSTM in order to predict daily sepsis, vancomycin antibiotic, myocardial infarction, and administration over two week patient intensive care unit (ICU) courses in the MIMIC-III dataset. They demonstrated that Attention improves the degree of interpretability to clinicians and makes DL approaches more flexible.
Choi et al. [62] have proposed a GRaph-based Attention Model, called GRAM for healthcare representation learning. The proposed model is different from what we have cited above. In fact, GRAM used medical ontologies and a RNN to model patient visits. First, it supplements EHR with hierarchical knowledge represented by medical ontologies. Based on the data volume and the ontology structure, GRAM represented a medical concept as a combination of its ancestors in the ontology via an attention mechanism. Unlike other methods, the medical concept representations learned by GRAM are well aligned with the medical ontology. GRAM exhibits intuitive attention behaviors by adaptively generalizing to higher level concepts when facing data insufficiency at the lower level concepts.

Interpretation based on rule extraction

Rule extraction from neural networks (NNs) is one of the techniques that allows to interpret and imitate the NNs behavior. However, there are only few applications that address this issue in the medical field.
Authors in [244] extracted rules from a MLP with small number connections and hidden unit activations. First, they removed the redundant connections and units from the model. Using a clustering algorithm, they sub-grouped the hidden unit activation values into a small number of clusters. Then based on the clustered hidden unit activation values, the network outputs a classification rules that involve the input attributes. Extracted rules from the breast cancer diagnosis problem achieve more than 95 % in term of accuracy rate on both the training dataset and the test dataset and explains the network outputs in terms of the input attributes of the data.
In [78], the authors have proposed a rule extraction algorithm from neural networks called ExTree. The algorithm has been trained on medical datasets for (diabetes, hepatitis, primary tumor, and heart datasets) for classification problems. ExTree can map and discover complex relationships between inputs and outputs. First, ExTree extracts a DT from the NN. Then, it generates from the decision tree an "if-then" rules.

3.2.3 Discussion

Recent works that treats EHRs for prediction tasks have some limitations.
First, they use either only personal features [26], [115] or clinical features [97]. The former discards a huge proportion of information in each patient's record, while the latter ignores knowledge and guidelines coming from human intelligence. Comorbidity features is another important information that can be considered for the clinical prediction. Comorbidities refer to the presence of one or more disorders that occur at the same time as a primary disease. This kind of information has an important implications for treatment and prevention. Previous studies have used different predictive variables, but to the best of our knowledge, analysis is lacking on the combined effect of the different types of factors.
Secondly, with the development of the ML, the explanation of the classification results has become a necessity, especially in healthcare. In fact, interpretability in ML models allow medical experts to make reasonable decisions and to provide personalized decisions that can ultimately lead to higher quality of service in healthcare. Not all applications-based ML techniques for healthcare make balance between performance and interpretability of the predictive model. The use of non-interpretable ("black box") techniques achieves a good prediction results but do not provide explanation behind prediction.
Thirdly, DL models are obtained from training large amount of data. This purely data-driven learning may induce contradictory results that can be uninterpretable. Injecting prior knowledge in DNNs is desirable to guide the learning step of models and reduce their non-interpretability. Until then, few presented applications were intended for the use of injecting prior knowledge.
Though, these studies have independently shown improved performance on various prediction tasks. To the best of our knowledge, there is no approach that addresses the limitations mentioned above simultaneously.
In this project, we proposed a medical support system that advances state of the art in several ways. First, our system integrates clinical data, personal data and comorbidity data and studies their impact according to the prediction task. Second, compared to the recent DL methods focusing on predicting medical events, our system can provide an understandable prediction to the decision maker. With this information about the rationale behind the model, the doctor will be empowered to trust the model or not. In addition, prior knowledge can be injected to avoid contradictory results and include medical knowledge in decision making process. Lastly but not least, unlike most of the existing methods, we can predict not only the optimal treatment but also the date of the next visit and then allow patient to be notified when necessary. So, the system can be intended for the use of not only for the medical expert, but also for the patients to look at their state of health and notify them of abnormal situations. In the literature we have not found enough works using rule extraction from DL for healthcare. This is what we will try

to propose in our medical support system to introduce interpretability for the user.

3.3 Multiagent systems and Argumentation in medicine

Several works have been carried out by reconciling MASs and medical field on the one hand and argumentation and the medical field on the other hand, more particularly in decision support systems. This will therefore be the subject of the following subsections.

3.3.1 Multiagent systems and healthcare

A large amount of works is interested in the properties of MASs meeting many needs of computer systems related to health. The medical world is well known for its complexity and an environment like the hospitals is, every day, the theater of an incalculable number of actions and interactions on the part of the actors (Patient, Doctor, Administration, etc.). Another important detail is that, despite the protocols and procedures put in place in the healthcare setting to standardize processes, unknown and human factors add a complexity and unpredictability to the system. The advantage of using agents, representing different actors of the environments, is that it allows to effectively represent and visualize the state of the system. In addition, the agent's autonomy [52] allows to observe the evolution of the state of the system without the need to intervene or to have domain knowledge. The objectives of MASs turn out to be particularly varied, which makes it adapted to various applications.
To offer a clear vision of the variety of MASs in medicine, we will take up, in part, the classification based on the medical sub-domains. Below, we cite the main sub-domains with brief description:

- Decision support systems: this category applies some type of data analysis techniques (for example ML) and also might often provide support to the decision maker;
- Data management systems: these systems focus mainly on health data representation, extraction, organization, storage, and presentation;
- Monitoring and assistive systems: includes systems designed for automated patient monitoring remotely, and patient self-care;
- Planning and resource management: the management focuses on planning medical processes, monitoring of staff and performance measurement, patient health monitoring, hospital, and clinical resources management;
- Privacy and security for healthcare applications: security and privacy are a priority in this kind of systems, given the sensitivity of the information being handled.

Authors in [137] [138] and [48] provide a detailed overview based on medical domain about the involvement of MAS in medical or health care domain.

Decision support systems

Decision support systems can take many forms but all aim to provide support medical to personnel in the various decision-making processes. Some systems are implemented to provide elements such as similar records of the patient to support physicians in the decision-making process such as MYCIN [269] and DXPLAIN [23]. MYCIN [269] is the first designed computer-based consultation system which aims to assist physicians in the diagnosis of the therapy selection for patients with bacterial infections. Moreover, MYCIN provide an explanation which can justify the provided advice or decision. DXPLAIN [23] is another decision-making systems developed at the hospital of Massachusettsis. It aims to predict diagnosis by considering disease signs, laboratories tests and the symptoms. The system was evaluated with a dataset containing more than 4500 clinical signs which are associated with more than 2000 different diseases.

Others rely on tools like ML techniques to provide answers and accelerate decision-making. The definition of the procedure to follow according to the place in the workflow of the patient and the doctor can also be subject to a decision support tool. Singh et al. [254] proposed a system called Healthcare Intelligent Assistant. The clinicians can use this system to solve medical cases. Based on the use of what authors call "case-based format" through agents, the user can capture the experiential knowledge of clinicians. Agents are made up of the past experiences of different clinicians. The final decision comes from aggregating the experiences of different clinicians into a large response for the user.
HealthAgents [116] is an agent-based system for diagnosing brain tumors. The "user's agent" communicates with "classifying agents" each located in different medical centers. The data is not identical in these centers. Therefore, the classifiers have a unique point of view on the offered case. "Classifying agents" are invoked to classify cases. Various factors that contribute to the classifier choice are used to rank the results. The practitioner only must interpret the decision and assesses the confidence associated with each of provided decision.
The main applications based MASs in the medical field were destinated to the doctors to support them in the decision-making process. For example, authors in [283] have introduced a model of multiagent diagnosis helping system (MADHS). The system uses four different kinds of agents: Coordinator, Specialists, Examiners, and Joint Decision Maker. It uses coordination and negotiation among agents to provide a diagnosis for a patient.

Data management systems

Data management systems aim to facilitate actions relating to data, such as preprocessing and retrieval. Due to the virtual and distributed nature of EHRs, the multiagent context is perfectly in accordance with this property. An example of such a system is the MAID system (MultiAgent System for Integration of Data) [75] developed by Cruz-Correia et al. The proposed system consists of several agents whose objective is to integrate the data provided by the various systems inherited from the hospital and aggregate them into an EHR. The files are stored in a central repository which can be accessed by users. These various agents constantly ensure that the information are made available to users is up to date. If an information requested by a user is not presented on the repository, then the agents organize one another to expedite the arrival of the information to the user. In another context, the CHIS (Context-aware Hospital Information System) [226] provides a ubiquitous system of information for the entire hospital. This makes it possible for practitioners and nurses to have access to basic services such as access and information sharing between staff members. But it is also possible, depending on the individual's authorization level, to observe the location of equipment and personnel through locator agents.

Monitoring and assistive systems

Monitoring and alarm systems are placed in a context where patients must be observed permanently, such as in geriatric services or with heart failure. Rather, these systems are integrated into hospital infrastructures, where staff cannot devote their time to observing a single patient. These systems are therefore based on the use of sensors, which continuously record information of the patient and identify any abnormalities.
Kafali et al. produced $COMMODITY_{12}$ system [145], which is dedicated to healthcare professionals for diabetic patients. $COMMODITY_{12}$ is based on patient physiological data that is collected through various portable devices. This information is then submitted to the judgment of expert agents, which analyze this information based on medical knowledge. Expert agents provide recommendations and warnings to the patient to inform him/her of his situation. These agents can also communicate with healthcare professionals to get help if needed. The care platform systems overlap some of its objectives with monitoring and alarm systems in that they often use sensors to perform diagnostics on the patient's state of health. However, the specificity of this approach compared to others lies in the fact that this system is placed in a very personal

context since it fits into the daily life at the patient's home.
Thus, agents promote telemedicine by allowing remote monitoring of the patient by the practitioner or by allowing the setting up of remote visits by physicians. Su et al. [257] apply this concept to the observation of distant fetal development. This approach allows the pregnant woman to be able to obtain information about the fetus when she wishes it without having to move (therefore to tire). This is also true for the obstetrician who can easily keep an eye on all patients.
In the same context of assistance at home, authors in [172] have proposed GAAMAAA (Generating Automatically an Adaptive Multiagent system for Ambient Assistive Applications), a system that aids suffering person maintained at home. The person should be equipped with a connected objects and may be including a robot. Each agent encapsulates information's coming from a connected object. The agents have been generated automatically. This makes their interaction dynamic. This work is an improvement of the COALAA project [1] where a personal assistance ontology includes a description of the installed connected objects as well as the knowledge about the person's profile.
Personal Agents (PAs) have been used as assistants to help users in their daily activities. Few works use PAs in the healthcare domain, where they can assist medical experts' activities and reduce medical errors. In this context, authors in [188] have proposed one of the first works that integrates personal agents and cognitive services for the prediction of the risk of shock in the next 15 minutes, and provides alerts in case of high risk to medical experts . The model enhances Belief-Desire-Intention agents reasoning with advanced cognitive capabilities to empower the reasoning capabilities of the agents and reduce medical errors. Authors have integrated the proposed model in the care path of trauma resuscitation, stepping forward to the Smart Hospitals.
Authors in [235] have proposed an Internet of Things (IoT) platform called AMBRO, which contains an intelligent cloud system layer. The platform allows to collect information about heart rate, the user/patient location and possible fall detection. Personal agents learn on data acquired by the system and act by sending notification alerts to caretakers.

Planning and resource management systems

Planning systems use agents to define and facilitate the emergence of an optimized organization of schedules and/or resources. The criteria for this optimization can be varied based on budgetary considerations, staff preferences, location of equipment, etc. CAMAP (Context-Aware MultiAgent Planning) developed by Ferrando et al. [102] is a system taking advantage of the autonomy and intelligence of the agents composing it. When an agent comes up with a plan, the agents of the whole system clearly articulate their opinions for judgment by their peers. Thus, the negotiation process begins. Each of the agents involved in the negotiation process is then free to present arguments or counterarguments against the proposed course of action. The most important capacity of the agents presented in this work is their capacity to give up and modify their behaviors by considering the context of the negotiation and arguments of other agents.
Simulation-oriented systems have also been used for providing tools that simulate the functioning of an environment. From these simulations, it is then possible to observe faults or isolate the causes of malfunction in the system. The objective is to somehow diagnose the simulation to improve the real situation it emulates. Silverman et al. [252] investigated such an approach to study the mental health and well-being of the people of Philadelphia. The agents of the simulation are defined by motivations and a state (including psychological, physiological, and socio-economic state) evolving over time, which influence the actions performed. The goal is to allow better allocation of hospital's resources for better patient monitoring and minimization of the risk of hospitalization. The results take the form of new instructions and work practices to achieve the objectives.
Simulating the spread of a virus in an emergency department allows to study different scenarios in detail and to be able to put in place appropriate infection control protocols. Laskowski et al. [159] underline the fact that the multiagent paradigm brings new elements in the understanding of how propagation works, notably with counter-intuitive facts that would not have been brought

to light in normal times.

Privacy and security for healthcare applications

Finally, the last aspect focuses on the implementation of secure systems. The evolution in a distributed paradigm indeed requires the presence of a certain number of constraints, especially in the medical world. Therefore, a system must always be available since a failure can have serious consequences for staff and patients. On the other hand, security and privacy must be a priority given the sensitivity of the information being handled. The multiagent paradigm naturally brings specific constraints such as the presence of malicious agents or the implementation of secure protocols for agent's communications.
Isern and Moreno [138] address the problem of the relatively small number of works focusing on the implementation of secure systems. Many systems are offered without guarantee from a safety point of view either because of technical constraint or by omission. Among this little state of the art, we will mention some recent application of Intelligent Multiagent Based Systems For E-Healthcare Security according to a recent study by Khan et al. [151].

Discussion

In recent years, MASs have become a growing subject of research to solve the limitations in the medical domain which is a huge environment distinguished by its common and distributed characteristics. In this subsection, we have shown how MASs have been incorporated in the medical field to solve various kinds of medical problems.
The present work belongs to decision support systems since our objective is to predict medical diagnoses and the date of the next visit. Decision support systems described above can be classified into three sub-categories: the patient-centered approach, the personal-centered approach, and the organization-centered approach. The first aims to provide tools and services directly to the patients, often with an emphasis on personalizing the care they receive. The second concerns systems that support staff in carrying out their daily tasks, often taking the form of a personal assistant. The third aims to provide the tools necessary for an organization to simulate and improve its functioning in general.
Unlike existing works, our framework will be intended for not only the use of medical expert but also for the patient because this latter needs to seek tools to look at his health and get notified in case of abnormal situations.
Thanks to their robustness, reliability and capacity for distributed processing, representation and reasoning, MASs are a powerful tool for distributed diagnostics. However, MAS need to use another discipline to obtain predictive analysis from data for a specific purpose. Decision support systems utilize knowledge base (KB) and apply some type of data analysis techniques such as ML, pattern recognition algorithms, and might often use knowledge inference techniques. For our case, we will use DL algorithms to exploit their effectiveness to handle with EHRs challenges and to provide good prediction results.
Our goal is to develop a multiagent architecture incorporating ML models to meet the needs of doctors and patients with Hypertension. The advantage of this approach is that it offers a system which can predict clinical events. For complicated tasks such as decision support, the integration of learning algorithms within the agents is an ideal solution to exploit the medical data of patients waiting to be used. This data contains important information on the profile of each patient and can be used to personalize their care. Our system would allow physicians to take this aspect into account and better adapt decisions to the profiles of patients.
To exploit the adaptation and reasoning distribution capacities of MASs to introduce intelligence into the way of using learning algorithms, we opted to use Argumentation. In recent years, many researches have focused on argumentation theory as a method of integrating knowledge into the field of data mining. Several approaches have been proposed to integrate the argumentation to MASs and learning algorithms. Unlike most of existing works which use argumentation as an

Approach	Knowledge base	Task	Arguments building
Hunter's framework	Inference rules from clinical guidelines	Representing and synthesizing clinical trials involving multiple outcome indicators	Agregation of three clinical guidelines
CONSULT	Logic rules	Supporting Patients in Self-Managing their Chronic Conditions	Official guidelines
Cyras's framework	Rule-based deductions	resolving conflicts among guideline recommendations	Combination of patient's EHR with clinical guideline
ABCN2	"if-then" Rules	Severe bacterial infections in geriatric population	Learning examples and prior knowledge
Wiliam's framework	Logic rules	Breast Cancer Prognosis	Relationships amongst variables and prior knowledge
ArgMed	Argumentation schemes	Analyse clinical discussions and justify the final decision	Exchange of views among healthcare professionals

Table 3.2: Some aproaches based argumentation properties

interaction protocol in MASs, the arguments of our system will be extracted automatically from DL algorithms. Based on prior knowledge, the extracted arguments will be evaluated so that the conflicting arguments will be rejected.
In the next subsection, we will present applications that have used argumentation for healthcare.

3.3.2 Argumentation and healthcare

Clinical reasoning is a complex phenomenon invariably defined in terms of the cognitive processes that healthcare professionals use to analyze and interpret a patient's medical information with reference to their prior knowledge and experience. Today clinical practices fall within the therapeutic and diagnostic fields and are defined in terms of diagnostic, management, and advisory skills. Diagnosis, management, and advice are each characterized by primary communication objectives. Therefore, arguments are used to generate the reasons that support the communicative goals associated with the essential skills of clinical practice [113].
Argumentation theory has been applied in the medical domain to make decisions with clear reasons supporting them based on the given data, the prior knowledge or clinical trials from guidelines.
In this thesis, we plan to design a support medical decision system which combines ML and argumentation. The system uses structured argumentation based on if then rules and is able to integrate prior knowledge that's why we are interested in three aspects:

- Argumentation based guidelines;
- Argumentation based schemes;
- Argumentation based ML;

Table 3.2 shows the characteristics of the some approaches based on argumentation in healthcare.

Argumentation based guidelines

In hospitals, national guidelines included in the Therapeutic Guide [198] are provided to medical professionals to provide the best care to patients. Guidelines include the procedures and the recommendations that deal with diverse clinical situations and guarantee the best evidence into best practice by improving diagnostic accuracy, promoting effective therapy, reducing healthcare variations, and discouraging ineffective interventions.
In the literature, a lot of works have used argumentation-based guidelines in healthcare. Authors in [135] have proposed argumentation-based techniques to aggregate the conclusions of various clinical trials and to determine which of two treatments is more effective according to a given situation. They have extracted evidence from clinical guidelines. Thus, arguments and counter-arguments were generated by inference rules for claiming that a treatment is superior to another based on the preference and available rules. Based on treatment indicators and the importance of evidence, arguments attack other arguments. To validate their framework, authors aggregate the evidence undertaken of three clinical guidelines involving 56 items of evidence and 16 treatments. For this purpose, they have used a structured argumentation formalism based preferences, semantics of grounded [88] and preferred extensions in order to identify the acceptable arguments.
Rather than determining treatment superiority based on clinical guidelines, Cyras et al. [77] focus on resolving conflicts among guideline recommendations when managing multimorbidity's. They have introduced a framework which considers patient's preferences to resolve recommendations conflicts. The system combines a patient's EHR with clinical guideline representation to obtain personalized recommendations. For this purpose, argumentation techniques-based preferences are used to resolve conflicts among recommendations. In addition, the framework can take feedbacks from the decision makers and integrate them. It also aims to explain decisions.
Grando et al. [121] focus on recommendations from a single guideline, rather than reasoning with conflicting recommendations from multiple guidelines. In fact, they have used statements in guideline as arguments then they aggregated confidence of arguments to identify the acceptable ones. The framework includes two kinds of mechanisms: the first one is an argumentation-based decision support system which represents medical decisions and chooses dynamically the most suitable plans to achieve medical goals. The second one consists in specifications related to the medical environment which can be considered before taking decisions. This improves the quality of care. As a case study, they chose to extract arguments from the hypertension guideline which defines the possible treatments to achieve medical goals based on the patient's condition.
Another work that applies argumentation for reasoning with guidelines and patient's preferences is CONSULT proposed by Kokciyan et al. [154]. CONSULT is a decision-support framework which aims to resolve conflicts among recommendations. In collaboration with healthcare professionals, CONSULT supports patients to self-manage chronic conditions and adhere to agreed-upon treatment plans. The advantage of CONSULT is that it considers the various preferences of the patient and the clinicians. Like the project presented in [135], Authors have extracted arguments from guidelines. Then, they analyzed arguments to resolve conflicts based on patient/clinician preferences. They also used argument schemes [273] and argumentation semantics to resolve inconsistencies among recommendations. At the end, CONSULT recommends the decisions which are the claims of the justified arguments.
Mayer et al. [175] have designed a system called ACTA to facilitate the work of clinicians in analyzing clinical trials. It consists in going beyond the basic keyword-based search in clinical trial abstracts, and it provides to the clinician the main linked claims and premises stated in the trial. Thus, structured "summary" of the abstract under the form of a graph is provided to the clinician rather than the whole abstract.

Argumentation based schemes

Argumentation scheme [223] [273] is a template that represents a common type of argument. According to Walton [273] argumentation schemes are in the form of premise-conclusion. Each argumentation scheme has a name, a set of premises, a conclusion, and a set of critical questions. Schemes allow to represent knowledge for arguing and explaining by capturing common patterns of reasoning. Argumentation schemes have been used in the medical field.

Works proposed in [4] [263] share the same idea of integrating argumentation with preferences to their support decision systems in order to allow clinicians to construct, exchange and evaluate arguments for and against decisions based on argumentation schemes.
Tolchinsky et al. [263] have used argumentation with preferences in a multiagent deliberation about organ transplantation. To construct arguments and attacks, expert clinicians use argumentation schemes from clinical guidelines. The arguments are then evaluated by a mediator agent to determine their strength. For this purpose, the mediator agent uses preferences over arguments based on the knowledge from clinical guidelines, the knowledge about past transplantations and the experiences of agent's interactions.
In order to detect conflicts between clinical discussions, the authors in [3], have proposed an approach based on argumentation schemes that analyzes clinical discussions with the aim of fostering the exchange of views among healthcare professionals. They opted for structured representation of reasoning patterns based on the argumentation schemes to interpret the assertions of the participants and to generate a graph of arguments which represents attack relations among them. This approach is based on one of the 25 schemes proposed by Walton [273]. The final graph is used to analyze clinical discussions and find out the conflicting ones or may be to add information.
Authors in [3] continued their study in the same subject to design ArgMed [4] which uses argumentation schemes to justify the final decision. In fact, ArgMed is an interactive system that supports decision making processes occurred during clinical discussions. Authors addressed the problem of not justifying the final decisions because clinical discussions are not documented, and only the final decision is recorded on patient electronic records. Therefore, the justifications for decisions made are not clarified. For this purpose, they suggested to represent discussions in a structured way, to formalize discussions based on a set of argumentation schemes that are considered valid in the medical field, then to identify invalid reasoning steps.
Atkinson et al. [18] have presented their model called DRAMA (Deliberative Reasoning with Arguments about Actions) which is an argumentation-based approach. Arguments are collected from various information sources. The model used argument schemes and multiple knowledge bases. To recommend a treatment based on safety and efficacy, values associated for each argument. Hence, treatments with higher values are recommended regarding a strict partial ordering on the values.

Argumentation based machine learning

The integration of argumentation and ML have been proven to be efficient. Existing approaches that have combined ML and argumentation differ in the way of integration and the medical sub-domain.
Authors in [277] have combined argumentation with Bayesian nets for Breast Cancer Prognosis. They represent knowledge by using logical arguments. Then, they built a Bayesian net by using the prior knowledge and a database. The Bayesian net can capture the probabilistic relationships amongst the features. In their case, argumentation theory has been used to aggregate clinical evidence as well as to provide a qualitative explanation of the prognosis. New arguments are generated based on Causal hypotheses gleaned from the Bayesian net which defines relationship attacks between arguments.
Authors in [286] proposed (ABCN2), inspired by Argument Based ML (ABML) as rule learning from examples which combines ML and argumentation. ABCN2, an argument-based extension of the CN2 rule learning [66], deals with severe bacterial infections in geriatric population. The aim of ABCN2 is to induct rules in argument-based framework for ML. Thus, the framework uses rules that cover the learning examples. Learning examples are guided by arguments for a better prediction result. The expert chooses a subset of learning examples and gives reasons behind the choice of the class for an example in form of arguments. The expert knowledge is passed to specific examples to justify the class. Several examples were generated by the medical doctor and used in the learning process.
In the literature, there is a few works that combine argumentation and ML in healthcare. However, argumentation and ML were combined in other domains. For example, in social media,

Habernal et al. [124] have used DL models to determine relations between arguments. Both bidirectional LSTMs (BiLSTMs) associated with an attention mechanism and a convolution layer over the input were used to determine why some arguments are more convincing than others for a given class.
Whereas they focus on determining convincingness, authors in [67] try to identify attack/support arguments' relationship between two texts. Each input text is fed into a LSTM model, then a vector representation of the text is produced. The two vectors are then merged using various techniques. The final resulting vector is embedded into a softmax classifier, which allows to predict the label for the relation between the two texts.

Discussion

All these works cited above show that the use of argumentation is very effective in the field of health and more particularly in the medical decision-making process. In most works reconciling argumentation and the medical field, arguments are constructed from whether guidelines or from prior knowledge. None of all of them have extracted the arguments automatically from EHR. In this thesis, we extract arguments automatically from classifiers and prior knowledge. To the best of our knowledge, there is no application that combines DL and argumentation for medical decision making. We propose to use argumentation as a strategy for combining DL classifiers. This allows introducing intelligence and interpretability into the system since an understandable decision is provided to the user. Our method differs from traditional methods by the fact that it can provide result explanation and integrate internal classification knowledge in base classifiers rather than only classification results. Moreover, our system allows adding new knowledge and gives the possibility to the doctor to detect and to remove contradictory knowledge.

3.4 Conclusion

This chapter allowed us to show the approaches which are related to our project in the literature either in ML, MAS or argumentation in healthcare. We have raised the issues to be resolved in each part. In the literature we have not found enough works that combines argumentation and ML for healthcare. This is what we will try to propose in the coming chapters.

Chapter 4

Medical support system

Contents

4.1 Introduction

Our goal is to develop a medical support system based on ML models and multi agents argumentation to meet the needs of doctors and patients. The integration of learning algorithms within the agents is an ideal solution to exploit the medical data of hypertension patients. This data contains important information about each patient's profile and can be used to personalize their prescription. Our system would allow the doctors to take this aspect into account and better adapt to the needs of patients. This chapter is organized as follows. We first present the motivation behind this work. Then, we describe the general architecture of the system. This part allows to obtain a precise vision of the whole approach in order to better discuss our choices regarding the direction of our work. We present the system by defining potential models that meet our needs, in order to define the first prototype. Once choices are defined, we evaluate our system using different sources and types of data (artificial data, public datasets, real data) at our disposal. Finally, we will discuss our contributions by comparing them to related works.

4.2 Motivation

Machine learnig (ML) models can be considered as the best way to produce an automatic personalisation in medecine and treat a large amount of health data which constitute Electronic Health Records (EHRs). We were interested in ensemble methods [196]. Ensemble methods improve ML results by combining different models. Most of research works has focused on the advantages of ensemble methods to improve the performance of algorithms. However, one of their major drawback is their lake of transparency, since no explanation of their decisions has been offered. With the development of the ML in sensitive fields, the explanation of classification results and the ability to introduce domain knowledge inside the learned model have become a necessity.
In this thesis, objectives were to add a transparency to deep ensemble method by using multiagents argumentation. Argumentation was used as an ensemble strategy for DL algorithms combination, which is more comprehensible and explicable than traditional ensemble method (such as voting). Meanwhile, by using argumentation, performance classification is improved, internal knowledge can be exploited and it is possible to inject recommendations.

4.3 Architecture

We devote this part to present our support decision system which predicts optimal treatment for each patient, provides result explanation, integrates internal classification knowledge in base classifiers and allows prior knowledge injection. The system proceeds in two main phases: Arguments extraction phase and Multiagent argumentation phase (see Figure 4.1).

4.3.1 Arguments extraction phase

The classifiers are built using bootstrap training samples which are generated from the training dataset. As in bagging ensemble method [46], a bootstrap sample is obtained by a random selection of examples with replacement from the original training dataset.

Deep multilayer network

Deep multilayer network (DMLP) is used as base classifier for the ensemble method. A DMLP consists of an input layer that receives input examples, hidden layers that are fully connected to the previous and the next layers and an output layer that provides the network outputs. In DMLP, activation functions are used to result outputs of real values, usually between 0 and 1 or between -1 and 1. This allows probability-based predictions or classification of items into multiple labels which consists in producing the probabilities of an input example belongs to each class.

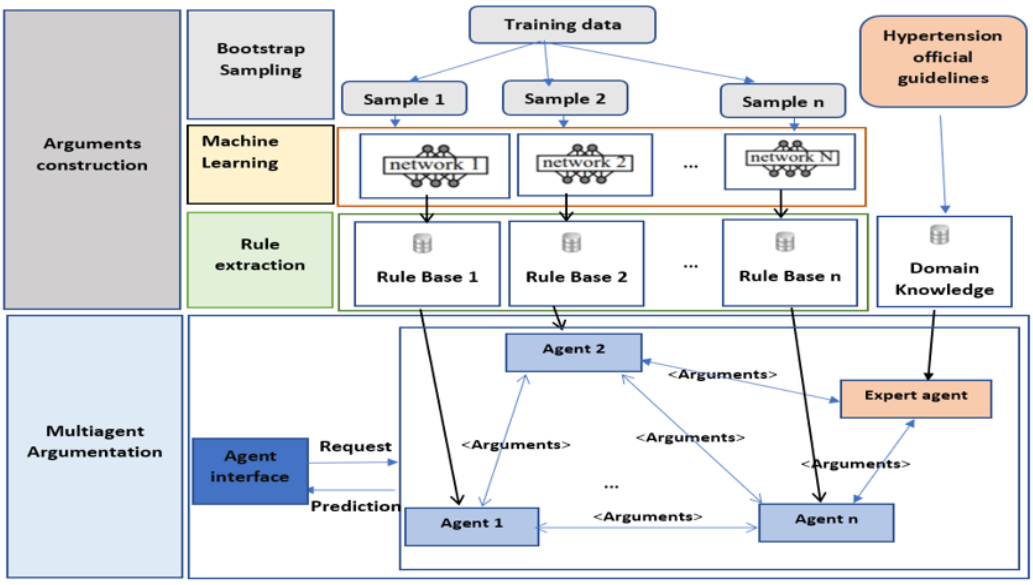

Figure 4.1: Medical support system architecture

Let's h_i^l the i^{th} neuron of the hidden layer l, its activation is defined by: $h_i^l = f(\sum_j w_{ji}^l h_j^{l-1})$, where w_{ji}^l is the weight of the connection from the j^{th} neuron of the layer $(l-1)$ to the i^{th} neuron of the layer l (h^0 represents the input layer) and f is the activation function. For the hidden layers, we used the Rectified Linear Units ($ReLU$) activation function, which gives good results in practice. It is defined as follows: $ReLU(x) = max(0, x)$. We used the softmax activation function for the output layer in order to obtain the probabilities of how likely the input $X = (x_1, x_2, ..., x_n)$ belongs to a class c. This function is defined by:

$$softmax(h_c^o) = \frac{exp^{h_c^o}}{\sum_l exp^{h_l^o}} \tag{4.1}$$

To train the DMLP, we used the adam [152] optimizer and the cross-entropy cost function L, which is one of the best choices in state-of-the art implementations. It is defined by:

$$L(Y, O) = -\frac{1}{N} \sum_i \sum_l y_{il} \ln (o_{il}) \tag{4.2}$$

Rules extraction step

Rules extraction step is very important since it allows to explain the predictions and to make the link between the classifiers and the MAS. To extract classification rules from DNNs, we have evaluated one pedagogical approach [74] and one eclectic approach [37]. We have chosen these approaches because they are scalable and adapted to the use of multiple DL algorithms. The extracted classification rules from each classifier constitute a rule base that is associated to the classifier. Each rule base is then embedded in an agent.
The form of a classification rule CR is:
$CR : (pr_1)\ (pr_i)\ ...\ (pr_n) \Longrightarrow (class(CR) = c, confidence\_score(CR) = s)$, where:
$pr_i \in premises(CR)\ (1 \leq i \leq n)$ are the premises of the rule CR that the example must satisfy to be classified in $c \in C$ (C is the set of classes). The form of the premise pr_i is defined by $pr_i = (x_i\ op\ \alpha_i)$ where x_i is the value of the i^{th} attribute, α_i is a real number and op is an operator. $s\ (0 \leq s \leq 1)$ is a confidence score that is associated to the rule CR. This score depends on the number $ne_c^+(CR)$ of examples that are well classified by the rule CR. To take into account the fact that most real datasets are unbalanced the number of well classified examples $ne_c^+(CR)$ is divided by the total number of examples ne_c in the class c : $confidence\_score(CR) = \frac{ne_c^+(CR)}{ne_c}$
Domain knowledge is also modeled in the form of rules, named expert rules (ERs):
$ER : (pr_1)\ (pr_i)\ ...\ (pr_n) \Longrightarrow (class(ER) = c)$, where $pr_i \in premises(ER)\ (1 \leq i \leq n)$ are the

premises of the rule ER that the example must satisfy to be classified in the class $c \in C$ based on the official experts' knowledge. For example, the ER_1 rule below expresses that an official recommendation for hypertension is to prefer the beta blockers (BB) treatment for young people: $ER_1 : (age < 50) \Longrightarrow (class(ER_1) = BB)$.
As said earlier, each rule base is encapsulated in an agent. In order to allow injecting prior knowledge in the system, an Expert agent is added for embedding the knowledge base which models prior knowledge provided by domain experts.

4.3.2 Multiagent argumentation phase

Modelling the argumentation process

Modelling the argumentation process consists in allowing each agent of the MAS to argue for its own prediction against other agents. So, we have focused on dialogical argumentation for the implementation of the argumentation process [222]. More precisely, agents engage in a process of persuasion dialogue [123] since they have to convince other agents that their prediction is better. Through the argumentation process, each agent uses the rules of its embedded rule base to answer to a prediction request and to provide arguments during the argumentation process. Since all the agents are able to participate to the argumentation process by exchanging messages, we have focused on multilateral argumentative dialogues protocols [40]. According to [176], multilateral argumentative dialogue protocol (MADP) is based on several generic rules that have been instanciated in our approach as explained hereafter. Seven communication performatives are used to instanciate the rules of the MADP as follows:

(a) Starting rules: the dialogue starts as soon as the user asks for a prediction. A_r uses the REQUEST performative to brodcast the request for a prediction. The content of the message is: $(X, ?c)$;

(b) Locution rules: an agent A_i sends an information by using the INFORM performative and asks for an information by using the ASK performative;

(c) Commitment rules: two rules are defined. The first one manages the prediction request by using the PROPOSE performative, allowing an agent A_i to propose an opinion by selecting the best rule that matches the request: $R_x^{i*} \in RB_x^i$ such that $confidence\_score(R_x^{i*}) = \max_{R_x^i \in RB_x^i}(confidence\_score(R_x^i))$, where $RB_x^i = \left\{R^i : R^i \in RB^i \wedge premises(R^i) \subset x\right\}$ (RB^i is the rule base associated to the agent A_i). The second rule allows an agent to declare its defeat by using the DEFEAT performative;

(d) Rules for combination of commitments: three rules for dealing with COUNTER, DISTINGUISH, CHECK performatives are defined. They define how acceptance or rejection of a given argument is performed. A_c uses the COUNTER speech act to attack the argument of A_m (associated to the rule R_x^{m*}) by selecting the rule R_x^{c*} such that $confidence\_score(R_x^{c*}) > confidence\_score(R_x^{m*})$. A_c uses the DISTINGUISH speech act to attack the opponent's argument, in case of equality of rule scores of A_c and A_m, they use the number of premises in their proposed rules as arguments: If $premise\_number(R_x^{c*}) > premise\_number(R_x^{m*})$ then A_c becomes the new Master ($premise\_number$ is the number premises of a rule). The expert agent A_e uses the CHECK speech act to check if the proposed rule R_x^{i*} by an agent A_i does not violate the rules $R_x^e \in RB^e$;

(e) Termination rules: the dialogue ends when no agent has a rule to trigger. The performative DEFEAT can be applied here to declare that the agent is defeated.

Moreover, it has been shown in [11] that agent role affect positively the argumentation process. So, in order to organize the dialogue, four distinct agent roles are defined:
(i) Referee: agent which broadcasts the prediction request and manages the argumentation process;

(ii) Master: agent that answers first to the Referee request;
(iii) Challenger: agent that challenges the Master by providing arguments;
(iv) Spectator: agent that does not participate to the argumentation process.
The Referee is an "artifact" agent role that is assigned in a static way. This agent interacts with the user for acquiring the prediction request and collecting the final result.
The argumentation process is performed through agent communication. For that purpose, we adopted speech acts language [241]. Let X be the input data, where X is a vector of attributes values $(x_i)_{i=1,...,n}$, c the class to predict. Three kinds of agent are present in the MAS: A_r the Referee Agent, A_e the Expert Agent which embeeds the ERs, the agents which embeed the CRs (A_m is the agent whose role is Master and A_c the agent whose role is Challenger).

Agents dialogues specification and behavior

The argumentation process begins as soon as the Referee Agent broadcasts a request for a prediction and manages the dialogue process. Each agent produces an opinion by selecting the best rule that matches the request. Once an agent sends its opinion, the Referee Agent sends its proposed opinion to the Expert Agent for verification. Expert Agent checks if the opinion matches with the recommendations, then it sends a message to the Referee Agent to express its acceptance if there is no conflict with the expert knowledge else it sends a rejection. The first agent which offers an accepted opinion becomes the Master. Other agents can challenge the Master by forming a challengers queue; the first participant in the queue is selected by the Referee agent to be a Challenger. All other agents except the Master and the Challenger agents adopt the Spectator role. For each discussed opinion, the agents can produce arguments from their individual knowledge base. When a Master is defeated by a Challenger, the Challenger becomes the new Master, and then can propose a new opinion. It should be noted that the defeated argument of the previous Master can not be used again, the previous Master can only produce a new argument to apply for Master once more. Otherwise, if a Challenger is defeated, the next participant in the Challengers queue is selected as the new Challenger, and the argumentation continues. If all challengers are defeated, the Master wins the argumentation and the Master's winning rule is considered as the prediction of the system. If there is no agent applying for the Master role, the argumentation is stopped. Since the number of arguments produced by the participants is finite and the defeated arguments can not be allowed to use repeatedly, the termination of the argumentation process is guaranteed. As we will see in the scenario illustration section, the output of the MAS contains not only the winning prediction and its explanation, but also the whole dialogue path which led to the result.

4.4 Experimental results

Ensemble Learning algorithms have advanced many fields and produced usable models that can improve productivity and efficiency. However, since we do not really know how they work, their use, specifically in medical problems is problematic. We illustrate here how our approach can help both physicians and patients to be more informed about the reasons of the prediction provided by the system. For this purpose, we evaluated our approach using artificial dataset, public datasets and real dataset.

4.4.1 Experimentation using artificial data

In this section, we describe our dataset and present the experimental results which allow to validate our approach using an artificial data.

Dataset description

We have used a specific dataset that is a realistic virtual population (RVP) [173] with the same age, sex and cardiovascular risk factors profile than the French population aged between 35 and

64 years old. It is based on official French demographic statistics and summarized data from representative observational studies. Moreover, a temporal list of visits is associated to each individual. For the current experiments, we have considered 40000 individuals monitored for hypertension during 10 visits per individual. Each visit contains: the Systolic Blood Pressure (SBP), the Diastolic Blood Pressure (DBP), class hypertension treatment, number of treatment changes etc. For hypertension treatment, 6 major classes of drugs have been considered: Alpha Blockers (AB), Calcium Antagonist (CA), Beta Blockers (BB), ACE Inhibitors (ICE), Diuritics (DI) and Sartans (SAR). The data of the RVP have been used to predict the treatment changing with our MAS, following the steps described in the precedent sections. In order to launch the experimentations, the MAS is built by encapsulating each rule base in an agent.

Scenarios illustration

We deal with an argumentative view of decision making, thus focusing on the issue of justification for the best decision to be made in each situation. Such an approach has indeed some obvious benefits. On the one hand, not only the best choice is suggested to the user, but also the reasons of this recommendation can be provided in an easy-to-understand format. On the other hand, such an approach to decision making can inject recommended knowledge given by a domain expert. Two scenarios illustrate the argumentation process: the first without domain knowledge injection and in the second, we have injected few medical recommendations.

(a) Scenario 1 without prior knowldge injection: here we use three DMLPs. The architecture of DMLPs was determined empirically. The retained architecture contains two hidden layers, it consists of 22 input neurons, 22 neurons in the first hidden layer, 20 in the second hidden layer, 6 output neurons (five neurons representing the drug classes and one neuron representing the patients with no treatment). The DMLPs was trained for 1500 epochs. Each DMLP generates is built using one bootstrap sample. We extracted knowledge bases from the three DMLPs using the eclectic rule extraction approach proposed in [37]. Extracting rules from neural networks allows to give an overview of the logic of the network and to improve, in some cases, the capacity of the network to generalize the acquired knowledge. Rules are very general structures that offer a form easy to understand when finding the right class for an example. Table 4.1 shows the properties of the three rule bases extracted from the three DMLPs in terms of number of rules per base, examples per rule, premises and premises per rule. For example the rule base RB^1 contains 182 extracted rules, it uses in total 96 different premises, the average number of premises per rule is about 7.7 and the average number of examples per rule is about 652.7. Each rule base is then embedded

Rule Bases properties	RB^1	RB^2	RB^3
Number of rules	182	351	256
Number of premises	96	106	88
Number of premises per rule	7.7	9.1	9.7
Number of examples per rule	652.7	752.9	395.4

Table 4.1: Rule bases properties.

in an agent. The rules are thus considered as individual knowledge of the agents. When a prediction is requested, instead of predicting the treatment class by majority voting like in classical ensemble methods, each agent uses a rule, which matches the request, to argue with other agents in the MAS in order to provide the best prediction for the current request. The process of argumentation executes as described in previous section. This is illustrated in the following example.
Let be p_1 a patient that is described by the following attributes: p_1:$[age = 64][sex = female][Visit_0 : SBP = 132.2,\ DBP = 79.5][Visit_1 : SBP = 125.3,\ DBP = 87.1][Visit_2 : SBP = 117.8,\ DBP = 89.1][Visit_3 : SBP = 103.4,\ DBP = 84.7]$.
p_1 should be treated by the treatment AC and the objective of the system is to predict this optimal treatment following the argumentation process described in section 2.1. The possible negotiation arguments are the weight of the rules. To simplify the current scenario, we

consider only the confidence scores of the rules. At the beginning of the scenario, the Referee Agent broadcasts the requested prediction, that is predicting the optimal treatment for the patient p_1. Then, each agent produces its opinion and asks for the Master role. Agent A_1 becomes the first Master and offers its opinion as follows: "this case should be in the class DI depending on the rule: $R^{1*}_{p_1}$: $(age > 54)(DBP_{Visit1} > 77.4)(DBP_{Visit2} > 85.1) \Longrightarrow (class(R^{1*}_{p_1}) = DI,\ confidence\_score(R^{1*}_{p_1}) = 0.61)$". Agent A_2 challenges agent A_1 using DistinguishRule as follows: "$R^{1*}_{p_1}$ is unreasonable because of rule $R^{2*}_{p_1}$:
$(age > 61)(DBP_{Visit0} > 142.5)(DBP_{Visit1} > 77.1)(DBP_{Visit1} > 77.1)(SBP_{Visit3} > 109.5) \Longrightarrow (class(R^{2*}_{p_1}) = BB,\ confidence\_score(R^{2*}_{p_1}) = 0.72)$".
The confidence score of the rule $R^{2*}_{p_1}$ is higher than the one of $R^{1*}_{p_1}$ Agent A_1 can not propose any rule to attack Agent A_2 and admits that it is defeated. Then Agent A_2 becomes the new Master and offers its own opinion.
The argumentation process continues until none of the agent is able to propose an opinion nor challenging another agent opinion. At the end, the master gives its prediction of the hypertension medication in a form easy to understand.
In this case, the final prediction is made by the agent A_3:
$R^{3*}_{p_1}$: $(age > 50)(DBP_{Visit0} < 81.3)(DBP_{Visit1} > 86.8)(SBP_{Visit2} < 120.3)(SBP_{Visit3} > 112.1) \Longrightarrow (class(R^{3*}_{p_1}) = AC,\ confidence\_score(R^{3*}_{p_1}) = 0.81)$.

(b) Scenario 2: with prior knowledge injection: injecting knowledge domain is very crucial for a decision making system. Medicine is one of the critical areas which needs the injection of recommendations for healthcare to improve the system reliability. In order to illustrate that, our approach improves the treatment prediction when adding domain knowledge.
We have injected few medical recommendations for hypertension treatment into the expert agent A_e. Examples of medical recommendations are given bellow: (age $<$ 50 years) $\Longrightarrow$ BB; (age $>$ 50 years) $\Longrightarrow$ DI.
The major role of A_e is to check if there is conflict between the proposed opinion and the expert knowledge. In this scenario, we have used three agents, each one contains extracted rule base from each DMLP and an extra agent which contains the expert knowledge. The process of argumentation executes as described in section 2.1.
This is illustrated in the following example.
Let be p_2 a patient that is described by the following attributes: p_2:$[age = 56][sex = female][Visit_0 : SBP = 112.2,\ DBP = 79.6][Visit_1 : SBP = 125.4,\ DBP = 89.7][Visit_2 : SBP = 103.7,\ DBP = 88.7][Visit_3 : SBP = 132.4,\ DBP = 81.7]$.
p_2 should be treated by the treatment BB and the objective of the system is to predict this optimal treatment as illustrated in Figure 4.2.

- At the first iteration T_1, the Referee Agent broadcasts the prediction request by transmitting the attributes p_2 and the requested class $?c$ to predict. Each agent produces its opinion by selecting the best rule that matches the request.
- At T_2 Agent A_2 proposes its opinion as follows: "the requested class should be ICE based on the rule: $R^{2*}_{p_2}$: $(age > 50)(SBP_{Visit1} > 101.4) \Longrightarrow (class(R^{2*}_{p_2}) = ICE,\ confidence\_score(R^{2*}_{p_2}) = 0.44)$".
- At T_3, the Referee Agent sends the proposed opinion of Agent A_2 to the Expert Agent A_e for verification in order to check if the opinion matches with the recommendations.
- At T_4, Expert Agent A_e sends a message to the Referee Agent to express its rejection and declares that $confidence\_score(R^{2*}_{p_2})$ is inapplicable since the predicted class DI (given by this rule) does not match with the predicted class of the recommendation rule: R^e_1: (age $>$ 50 years) $\Longrightarrow$ (DI).
- At T_5, Agent A_3 proposes its opinion as follows: "the requested class should be BB based on the rule: $R^{3*}_{p_2}$: $(age > 50)(DBP_{Visit2} > 80.9)(SBP_{Visit3} < 145) \Longrightarrow (class(R^{3*}_{p_2}) = BB,\ confidence\_score(R^{3*}_{p_2}) = 0.56)$".
- At T_6, Referee Agent sends the suggested choice to the Expert agent for verification.
- A_e declares that this rule is applicable since there is no conflict at T_7.

- At T_8, Referee Agent declares by an INFORM message that agent A_3 is defined as a Master.

- At T_9, Agent A_1 proposes its opinion as follows:"the requested class should be BB based on the rule: $R^{1*}_{p_2}$: $(age < 66))(SBP_{Visit3} < 135.1)(DBP_{Visit3} > 79) \Longrightarrow (class(R^{1*}_{p_2}) = BB,\ confidence\_score(R^{1*}_{p_2}) = 0.6)$".
- At T_{10}, A_e declares that this rule is applicable since there is no conflict.
- At T_{11}, Referee Agent declares that Agent A_1 is the first Challenger, Agent A_2 is Spectator. Since a Master and a Challenger are defined, the encounter arguments can be performed.
- At T_{12}, Agent A_1 (Challenger) asks Agent A_3 (Master) for its arguments in order to compare them with its own arguments.
- At T_{13}, Agent A_3 sends its arguments to Agent A_1. In this case, the score of the rule $R^{1*}_{p_2}$ (Agent A_1) is higher than the score of $R^{3*}_{p_2}$ (Agent A_2).
- Thus, Agent A_3 admits its defeat and Agent A_1 becomes the new Master and can propose its own opinion at T_{14}.

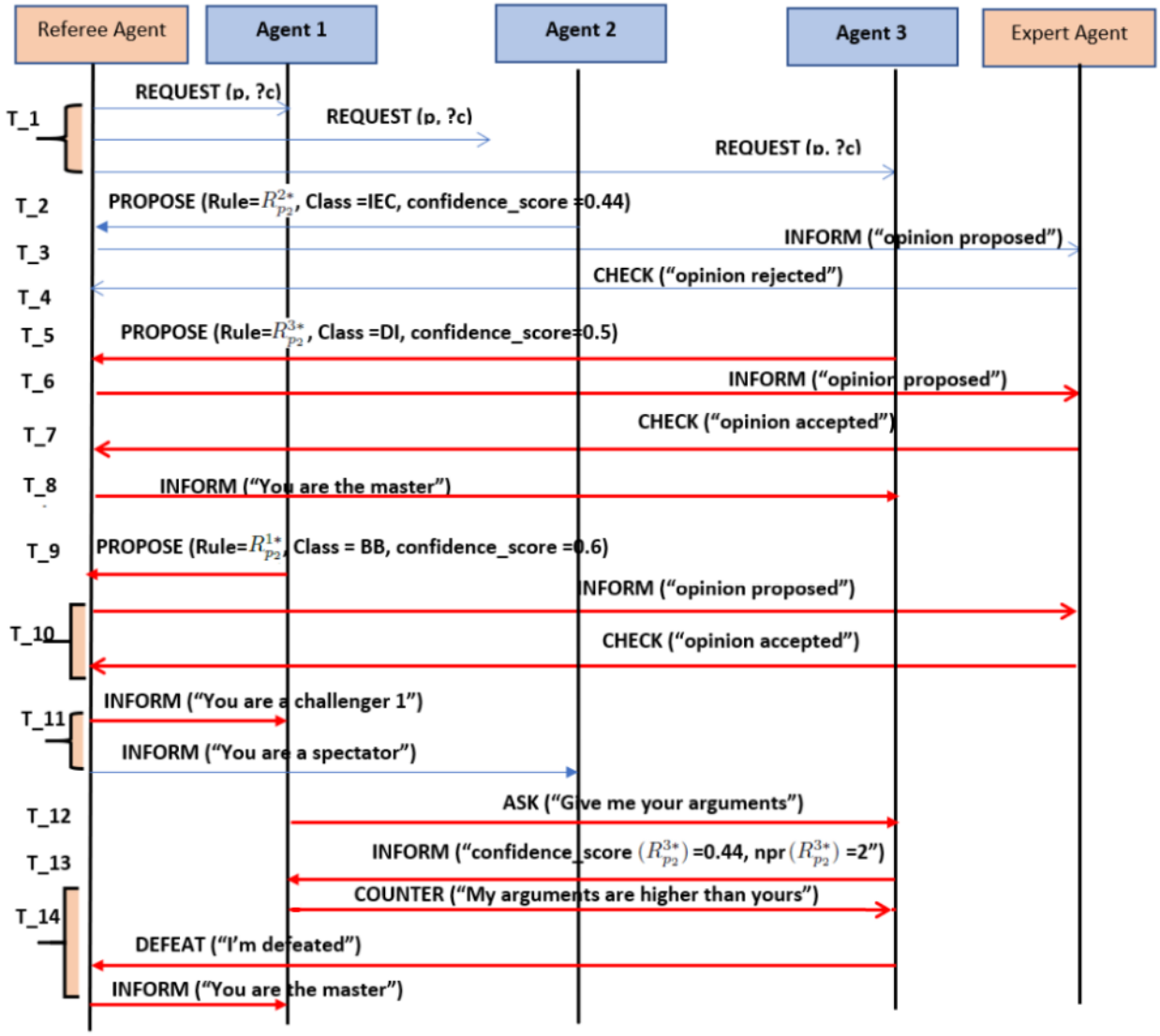

Figure 4.2: Illustration of the case study argumentation process.

The argumentation process continues until none of the agents is able to propose an opinion nor challenging another agent opinion. In case of equality of the confidence scores, the number of premises of the two rules are compared and the agent that have the highest one wins

the argumentation process. At the end, the final master gives its prediction of the Hypertension medication in the form of a rule which is easy to understand. The patient p_2 has been well classified and the system recommends him/her to take BB treatment based on the rule of Agent A_1: $R^{1*}_{p_2}$: $(age > 50)(DBP_{Visit2} > 70.0)(SBP_{Visit2} > 112.0)(SBP_{Visit3} > 130.5) \Longrightarrow (class(R^{1*}_{p_2}) = BB,\ confidence\_score(R^{1*}_{p_2}) = 0.72)$.

Performance prediction

We compared the two variants described above to: (i) the most popular ensemble learning methods (Bagging [46], AdaBoost [106]), XGBoost [56] and (ii) two classification approaches based on ensemble rule extraction that uses the DIMPL [38]: one trained by bagging (DIMLP-B) and another trained by arcing (DIMLP-A). In the argumentation process, we have used CRs and the provided ERs injected into the expert agent. The number of the bootstrap samples used in all the approaches is shown in Table 4.8. For DIMLP ensembles, we have used the default parameters defined in [38] (the number of bootstrap samples is equal to 25). Table 4.2 shows

Adaboost	Bagging	XGBoost	DIMLP-A	DIMLP-B	App1_DIMLP		App2_TREPAN	
Accuracy	Accuracy	Accuracy	Accuracy	Accuracy	Accuracy	Fidelity	Accuracy	Fidelity
79.0±0,05 (100)	76.0±0.04 (100)	78.1±0.01 (300)	85±0.09 (25)	79.0±0.02 (25)	**89.0**±0.02 (10)	**97.6**±0.01	79.1±0.01 (10)	91.1±0.07

Table 4.2: Results comparison to ensemble methods.

that App1_DIMLP gives better accuracy for the classification task than other ensemble methods. With the exception of DIMLP-A, App2_TREPAN outperforms other ensemble methods. Our method as an ensemble method can effectively reduce the error regarding to a single DMLP. Table 4.3 shows that our method (using 10 DMLPs) outperforms a single DMLP in two cases: when injecting prior knowledge and without injecting prior knowledge. As we can see in the

Single DMLP	without prior knowledge injection	with prior knowledge injection
79.8±0.01	83.2 ±0.03	89.0±0.01

Table 4.3: Comparaison to a single DMLP.

Table 4.3, expert knowledge injection improves accuracy of classification. We improved the results classification and explained decision by providing not only a comprehensible classification rule behind the decision but also the sent and received messages amoung the agents. So one can obtain a trace allowing to distinguish the unfolding communication between agents. In Figure 4.2, the red arrows show the messages that lead to the final prediction treatment for p_2 patient. Moreover, our approach is able to exploit domain knowledge that controls the system and gives trust to the expert.

4.4.2 Experimentations using public datasets

In the experiments we used 11 datasets representing classification problems. Table 4.4 illustrates their main characteristics in terms of number of samples and number of input features. The public source of the datasets is UCI: ML Repository at the University of California,Irvine: https://archive.ics.uci.edu/ml/datasets.html. Our experiments are based on 10 repetitions of 10-fold cross-validation trials. Training sets were normalized using Gaussian normalization. We compared three variants of our approach: (i) App1_DIMLP that uses the eclectic rule extraction algorithm described in [37]; (ii) App2_TREPAN that uses the pedagogical rule extraction algorithm described in [72] and (iii) App3_Extract that replaces the DMLPs and the rule extraction step by a rule extraction algorithm that extracts rules directly from the bootstrap samples.

Dataset	Nb Attributes	Nb Instances
Breast Cancer Prognastic	33	194
Bupa Liver Disorders	6	345
Glass	9	163
Haberman	3	306
Heart Disease	13	270
ILPD (Liver)	10	583
Pima Indians	8	768
Saheart	9	462
Sonar	60	208
Spect Heart	22	267
Vertebral Column	6	310

Table 4.4: Datasets chracteristics

In order to validate the performance of our approach, we compared the three variants described above to: (a) the most popular ensemble learning methods (Bagging [46], AdaBoost [106]) and (b) two classification approaches based on ensemble rule extraction that uses the DIMPL [38]: one trained by bagging (DIMLP-B) and another trained by arcing (DIMLP-A). We defined a grid search to optimize the parameters of each approach. The number of the bootstrap samples used in all the approaches is shown in Table 4.5. For DIMLP ensembles, we have used the default parameters defined in [38] (for example, the number of bootstrap samples is equal to 25). In the argumentation process, we have only used CRs because there are no ERs provided for the used public datasets. Therefore the argumentation process takes place without any expert agent. In the experiment, we have used the Accuracy and the Fidelity as evaluation measures to compare the classification performance of the different methods described above. Table 4.5 presents the

Datasets	Adaboost	Bagging	DIMLP-B	DIMLP-A	App3_Extract	App1_DIMLP		App2_TREPAN	
	Accuracy	Accuracy	Accuracy	Accuracy	Accuracy	Accuracy	Fidelity	Accuracy	Fidelity
Breast Cancer Prognastic	82.5 ±0.05 (150)	79.1 ±0.03 (125)	74.4 ±0.02 (25)	73.7 ±0.04 (25)	71.1 ±0.01 (12)	**88.7** ±0.03 (10)	98.8 ±0.03	84.3 ±0.07 (11)	97.9 ±0.09
Bupa Liver Disorders	83.2 ±0.09 (125)	78.0 ±0.21 (100)	67.3 ±0.08 (25)	61.9 ±0.03 (25)	75.8 ±0.03 (11)	**87.1** ±0.02 (10)	96.6 ±0.01	83.6 ±0.10 (10)	97.3 ±0.08
Glass	81.5 ±0.04 (100)	79.0 ±0.03 (100)	74.1 ±0.03 (25)	**81.9** ±0.06 (25)	76.9 ±0.08 (12)	79.9 ±0.10 (11)	97.5 ±0.01	81.3 ±0.06 (10)	96.3 ±0.06
Haberman	74.6 ±0.05 (100)	72.0 ±0.01 (125)	76.4 ±0.08 (25)	74.3 ±0.09 (25)	72.4 ±0.04 (12)	81.4±0.02 (10)	97.8 ±0.03	**83.7** ±0.05 (10)	97.9 ±0.01
Heart Disease	86.3 ±0.06 (100)	86.0 ±0.09 (100)	84.9 ±0.05 (25)	81.3 ±0.07 (25)	83.1 ±0.10 (12)	**86.6** ±0.01 (25)	97.1 ±0.01	77.1 ±0.03 (11)	97.0 ±0.03
ILPD (Liver)	73.4 ±0.09 (150)	71.1 ±0.01 (125)	69.3±0.02 (25)	70.2±0.05 (25)	70.0 ±0.06 (12)	**79.1**±0.01 (11)	96.9 ±0.01	74.9 ±0.03 (10)	95.8 ±0.07
Pima Indians	78.1 ±0.09 (100)	77.8 ±0.06 (100)	77.4 ±0.06 (25)	76.1 ±0.04 (25)	77.2 ±0.01 (12)	**80.9** ±0.02 (9)	97.8 ±0.07	77.6 ±0.01 (12)	96.9 ±0.01
Saheart	72.1 ±0.11 (150)	72.3 ±0.12 (100)	72.3 ±0.02 (25)	70.6±0.04 (25)	71.3 ±0.02 (9)	**74.8** ±0.09 (11)	97.1 ±0.03	72.1 ±0.13 (12)	95.7 ±0.03
Sonar	72.4 ±0.01 (100)	70.6 ±0.01 (100)	71.1 ±0.06 (25)	74.3 ±0.06 (25)	71.0 ±0.05 (11)	76.6 ±0.04 (10)	96.9 ±0.01	**79.9** ±0.06 (9)	96.7 ±0.06
Spect Heart	71.9 ±0.03 (125)	72.3 ±0.06 (150)	72.9 ±0.01 (25)	70.9 ±0.02 (25)	72.9 ±0.02 (11)	79.7 ±0.02 (9)	96.9 ±0.01	**81.9** ±0.01 (12)	96.7 ±0.03
Vertebral Column	74.9 ±0.03 (125)	72.3 ±0.01 (150)	82.9 ±0.03 (25)	81.1 ±0.05 (25)	80.6 ±0.03 (12)	**86.8** ±0.04 (11)	96.9 ±0.02	77.1 ±0.03 (10)	95.9 ±0.02

Table 4.5: Accuracy of ensemble methods.

obtained experimental results. It shows that our framework can effectively ensure high accuracy

results for the classification task on several datasets. In contrast with traditional ensemble methods, we can find that App1_DIMLP and App2_TREPAN outperform Bagging and AdaBoost methods using fewer classifiers. For example in Vertebral Column dataset, App1_DIMLP obtains an accuracy of 86.8% (using 11 classifiers) while the accuracy of Bagging and AdaBoost are lower than 75% (using more than 125 classifiers). Our method gives better results than DIMLP-B and DIMLP-A on the majority of datasets. For example, in Bupa Liver Desorders dataset, the accuracy of App1_DIMLP exceeds that of DIMLP-A by 25.2%. In Breast Cancer Prognostic dataset, the accuracy of App1_DIMLP is 88.7% (using 10 classifiers) while the accuracies of DIMLP-B and DIMLP-A are lower than 75%.
So far, the results have been in our favor for the predictive accuracy of the 10 out of 11 classification problems. Moreover, we can see that the Fidelity score is higher than 95% in all datasets. This means that the classification rules extracted from the DMLPs matches the classification results provided by the DMLPs.
Experimental results show that App1_DIMLP and App2_TREPAN give better accuracy for the classification task than other ensemble methods. Indeed the use of argumentation process allows to outperform the classical ensemble methods and also the rules extracted from ensembles. In addition, we have shown that using DL with rule extraction step gives better results than using a rule extraction algorithm directly from the bootstrap samples (App3_Extract). As a conclusion, we can deduce that App1_DIMLP and App2_TREPAN can effectively extract high quality knowledge for ensemble classifier and ensure high accuracy in classification as well. Moreover our method provides explanations and transparency of the predictions. It is able to extract useful knowledge from ensemble classifiers.

4.4.3 Experimentations using EHR

In this section, we describe our dataset and present the experimental results which allow to validate our approach using real data.

Dataset description

In the experiments we used a real dataset provided by CGEDIM (medical prescription platform used by 23000 doctors in France), which has been collected from 3000 doctors (see table 4.6). This dataset describes the characteristics of 429087 patients, each patient is represented by a series of visits. The average number of visits is 20 per patient. Each visit contains about 15 characteristics (see table 4.7). A class of treatment is associated to each patient at each visit, we distinguish 6 classes of treatment: Alpha blockers (AB), Beta-blockers(BB), Diuretics (D), Angiotensin II receptor blockers(ARAII), ACE inhibitors (IEC) and Calcium channel blockers (IC).
The dataset was anonymized by removing all identifiable features such as names, addresses and telephone numbers.

Number of patients	429087
Number of input features	15
Average number of visits	20
Type of features	real and boolean
Gender distribution (female,male)	41%, 59%
Average age at the time of prescription	68.8
Average duration between visits (days)	63.8
Treatment classes	BB, AB, IEC, ARAII, IC, D

Table 4.6: Datasets properties

Features	Signification
Gender	Gender
Age_presc	Age of patient at time of prescription
Weight	Patient's last recorded weight measurement
Height	Patient's last recorded height measurement
Box	Number of boxes prescribed to the patient
Quantity	Dosage expressed in number of doses (to be associated with the frequency_label field)
Frequency_label	Frequency of taking medication
Duration	Duration of the prescription expressed in days
Pulse	Heart rate associated with medical consultation
Diastolic pressure	Diastolic pressure associated with the medical consultation
Systolic pressure	Systolic pressure associated with the medical consultation
Prescription_blood_sugar	Patient's Blood glucose measurement associated with a medical consultation
last_mesure_blood_sugar	Patient's last blood glucose measurement
Insulines_treatment	If there has been an insulin treatment
Other_treatment	If there has been an A10 treatment (other than insulins)

Table 4.7: Features signification

Prediction results

In these experimentations, we have used the same protocol defined in the section 4.4.1. Unlike previous datasets, this one is a multivariate time series. A common way to perform prediction task on multivariate time series, is to use information of the previous visit to predict the next one using DMLP. The network is trained on a temporal window of inputs describing a fixed set of recent past states. Here, we employed DMLPs for time series clasification by using the slinding window technique in order to predict the optimal treatment for each patient. To avoid overfitting, we used Dropout between DMLP layers. The architecture of DMLPs was determined empirically. The retained architecture contains two hidden layers. The DMLPs was trained for 1500 epochs. We compared two variants of our approach: (i) App1_DIMLP that uses the eclectic rule extraction algorithm described in [37] and (ii) App2_TREPAN that uses the pedagogical rule extraction algorithm described in [72].
In order to validate the performance of our approach, we compared the two variants described above to: (i) the most popular ensemble learning methods (Bagging [46], AdaBoost [106]), XGBoost [56] and (ii) two classification approaches based on ensemble rule extraction that uses the DIMPL [38]: one trained by bagging (DIMLP-B) and another trained by arcing (DIMLP-A). In the argumentation process, we have only used CRs because there are no ERs provided for the used dataset. Therefore the argumentation process takes place without any expert agent. The number of the bootstrap samples used in all the approaches is shown in Table 4.8. For DIMLP ensembles, we have used the default parameters defined in [38] (the number of bootstrap samples is equal to 25). Table 4.8 show that App1_DIMLP gives better accuracy for the classification task than other ensemble methods. With the exception of Adaboost, App2_TREPAN outperforms other ensemble methods. Indeed, we can say that the use of argumentation as a method of combining classifiers can ensure high accuracy prediction with real data. This allows to validate our appraoch and make its use as a decision support system possible in hospitals. Moreover, we can see that the Fidelity score is higher than 94% with App1_DIMLP and App2_TREPAN.

Adaboost	Bagging	XGBoost	DIMLP-A	DIMLP-B	App1_DIMLP		App2_TREPAN	
Accuracy	Accuracy	Accuracy	Accuracy	Accuracy	Accuracy	Fidelity	Accuracy	Fidelity
77.2±0.13 (100)	72.7±0.06 (100)	70.1±0.03 (300)	64.4±0.08 (25)	63.7±0.16 (25)	**73.5**±0.03 (10)	**94.7**±0.04	67.4±0.08 (10)	94.3±0.17

Table 4.8: Results comparison to ensemble methods.

This means that the classification rules extracted from the DMLPs matches the classification results provided by the DMLPs.
Some extracted rules from a trained DMLP are presented below:

- Rule 1: (Age_presc > 59) (Box > 4) (Diastolic pressure < 93) (Systolic pressure > 193) Class = ARAII (789)
- Rule 2: (Pulse < 60) (Box > 4) (Height < 170) (Duration > 84) Class = D (810)
- Rule 3: (Systolic pressure > 140) (frequency_label < 2) (Diastolic pressure < 90) (Frequency_label > 140) (Prescription_blood_sugar > 5.46) Class = AB (456)
- Rule 4: (Weight > 87) (Box < 2) (Prescription_blood_sugar < 85) (Systolic pressure > 125) Class = BB (955)
- Rule 5: (Age_presc < 89) (Box < 4) (Prescription_blood_sugar < 5,46) (Systolic pressure > 140) Class = IEC (733)
- Rule 6: (Prescription_blood_sugar > 5,11) (Quantity < 2) (Systolic pressure > 140) Class = IC (841)

We have also compared the results of some existing approaches (Doctor AI, Med2vect, Deepr, DeepCare, DeepPatient) with our approach for predicting the optimal treatment.

Approachs	Accuracy	Precision	Recall	F1-Measure
App1_DIMLP	**73.5** ±0.03	72.2±0.13	76.2±0.09	74.1±0.01
App2_TREPAN	67.4 ±0.08	66.3±0.08	70.4±0.11	68.3±0.03
Doctor AI	68.3 ±0.07	69.5±0.01	65.2±0.02	67.3±0.03
M2vect	65.4 ±0.02	59.5±0.07	69.2±0.11	64.0±0.05
DeepCare	55.5±0.11	54.6±0.03	64.9±0.01	59.3±0.09
Deepr	51.9 ±0.04	52.2±0.02	46.6±0.03	49.2±0.02
DeepPatients	58.8 ±0.01	57.2±0.03	69.3±0.01	62.7±0.13

Table 4.9: Accuracy of algorithms for predicting optimal treatment.

Table 4.9 confirms that our proposed approach is able to outperform the existing ones by a large margin while predicting optimal treatment. App1_DIMLP obtains an accuracy of up to 73.5%, App2_TREPAN obtains an accuracy of up to 67.4% while the rest of the approaches give an accuracy lower than 66% except Doctor AI which obtains about 68.3% of accuracy. This can be explained by the use of deep ensemble method against one model in others. In fact, in theory our ensemble method improves ML results by combining multiple models using argumentation which is better compared to a single model.
App1_DIMLP and App2_TREPAN are able to extract useful knowledge from ensemble classifiers.
Moreover our method provides explanations and transparency of the predictions.

4.5 Discussion

Few works have previousely adressed rule extraction from ensembles. The DIMLP was used to extract rules from network ensembles [38]. Zhou et al. proposed the REFNE algorithm (Rule

Extraction from Neural Network Ensemble) [294], that extracts rules from instances generated from the trained ensembles. Hayashi et al. extended the "Recursive-Rule eXtraction" (Re-RX) algorithm to multiple MLP Ensemble [128]. None of these works used argumentation for performing predictions nor addressed the problem of knowledge injection into the algorithm. Over the last decades, argumentation has come to be increasingly central as a core study within AI since it attracts much attention in a lot of fields, especially ML. Existing approaches differ in their use of argumentation and in their choice of argumentation framework and method.
Zhiyong Hao et al. [125] present Arguing Prism, an argumentation based approach for collaborative classification which integrates the ideas from modular classification inductive rules learning and multiagent dialogue. Each participant agent has its own local repository (data instances) and produces reasons for or against certain classifications by inducing rules from their own datasets. The agents use argumentation to let classifiers, learned from distributed data repositories, reaching a consensus rather than voting mechanisms. This approach is interesting because it allows avoiding simple voting and generates arguments in a dynamic way. Unfortunately its use is restricted to DTs.
Bratko et al. [42] present a novel approach to ML, called argumentation based ML, which combines ML from examples with concepts from the field of argumentation. The idea is to provide expert's arguments, or reasons, for some of the learning examples. Reasons (arguments) impose constraints over the space of possible hypotheses, thus reducing search complexity and an induced theory make more sense to an expert as it has to be consistent with the given arguments. A part of this knowledge, consisting in an expert knowledge, is handly introduced into the system.
Maya Wardeh et al. [275] present a classification approach using a MAS founded on an argumentation from experience. Thus, a group of agents argues about the classification of a given case according to their experience which is recorded in individual local data sets. The arguments are constructed dynamically using classification association rule mining [2] techniques. Even if this approach argues for the use of local data for the argument exchange between the agents, there is a chairperson agent which acts as a mediator agent for the coordination of the whole MAS. From our point of view, this is is a weak point, since the system fails to perform any classification if the chaiperson agent fails. Different approaches achieve different and desirable outcomes, ranging from improving performances (reduce the combinatory search among possible hypotheses) to rendering the ML process more transparent by improving its explanatory power. All of these works illustrate the importance of building arguments for explaining ML examples. But all of them are dedicated to rule association [275] [281] or to DTs [125]. Since the existing approaches are built in a monolithic way (i.e. based on a monolithic algorithm), they lack robustness. If the algorithm fails, the whole system fails. In contrast, our approach consists in distributing the argumentation process through agents where embedded rule-bases act in autonomic way while argumenting with each other. Finally, the most important point is that, none of the existing approaches that combine ML and argumentation addresses DL methods, despite these are among the most powerful ML algorithms. The use of argumentation techniques allows to obtain classifiers, which can explain their decisions, and therefore addresses the recent need for explainable AI: classifications are accompanied by a dialectical analysis showing why arguments for the conclusion are preferred to counterarguments. Our method differs from traditional ensemble methods by the fact that it can provide result explanation and integrates internal classification knowledge in base classifiers rather than only classification results. Moreover, as shown by the experimental results, our method improves the predictions. Our argumentation protocol has been inspired from several existing argumentation frameworks. We have adopted structured arguments in order to deal with complex arguments and to take into account the confidence degree of the rules. The value-based argumentation frameworks allowed us to implement the confidence degree to quantify the strength of the arguments.
The preference and defeat relations among arguments have been defined in a simple way; by a ordinal relation. It will be easy to extend this by considering a deeper analysis of the premises of the rules.

Chapter 5

MS-LSTM: Multisources LSTM based attention

Contents

5.1 Introduction

Predicting the risk of potential diseases based on large sequences of patient's visits has attracted considerable attention in recent years, especially with the development of deep learning techniques wich can easily handle with a large volume of data. Compared with traditional ML models, deep learning based approaches [118] achieve superior performance on risk prediction task. In fact, recently the Deep Learning methods have shown that neural network models have brought great hope in health care research for drug discovery, treatment innovation, personalized medicine, and optimal patient care and improve patient outcomes. EHR contains the patient records and provides a long-term view of a patient health. This sequential data, including different information about diagnoses, demographics, procedures, and medications are represented by a high dimensional clinical variable and sequenced by patient medical visits. Predicting the future treatments or the future clinical events based on patient EHR is a critical task since each medical information may have varying importance. The most challenging issues in medical prediction tasks is the fact of correctly modeling the temporal and high dimensional of EHR data, interpreting the importance of medical features and information and improving the prediction results. As the number and the volume of temporal datasets increase rapidly, traditional ML algorithms are becoming overwhelmed. Recently, with the advances of deep learning techniques, deep leaning models such as RNNs [180][249] have enjoyed considerable success in various ML tasks due to their powerful hierarchical feature learning ability in modeling sequential data, and have been widely applied in various temporal data mining tasks such as predictive learning, representation learning and classification.
However, RNNs have limitations, for example it is difficult to train these networks on long input sequences. They cannot handle long sequences effectively. This is due to the problems of vanishing and exploding gradients that occur when errors are backpropagated across many time steps. LSTM [249] solved this problem by integrating memory units to enable learning of long temporal dynamics. Another limitation of using RNNs is the lack of interpretability. Interpretability is very important in the healthcare domain since it can lead to the design of suitable intervention mechanisms. To model the temporal EHR data and interpret the prediction results simultaneously, attention-based neural networks can be employed, which aim to learn the relevance of the data samples to the task. Attention is one of the most powerful concepts in the deep learning field. It is based on the intuition that, when processing a large amount of information, we focus on a certain part of this information.

In this section, we propose a novel deep architecture called MS-LSTM which uses LSTM and attention mechanism for medical prediction tasks. Multisources LSTM (MS-LSTM) can successfully combine several sources of medical data, consider the temporal trajectory of events embedded in EHRs, and identify relevant clinical factors that contribute to the prediction. In addition, the experiments are performed on real EHR data which will support experts in healthcare. Such method could result significantly better diagnoses and therefore better clinical outcomes. Moreover, results show a significant improvement in prediction results over existing state of the art expert systems.
In the following sections, we describe our method and show experiment protocol and results.

5.2 Architecture

We present a new temporal deep neural network architecture, called MS-LSTM, that is based on LSTM. It predicts both optimal treatment and the date of the next visit of a patient.
We have chosen LSTM because it is adapted to time series prediction due to its ability to remember previous inputs. Traditional recurrent neural networks suffer from two problems: vanishing gradient and exploding gradient, which make them unusable. LSTM can solve these issues by explicitly introducing a memory unit, called the cell, into the network.
MS-LSTM successfully combines different sources of medical information and considers the temporal trajectory of events embedded in EHRs. The overall approach is illustrated in Figure 5.1. Each information source is connected to multiple fully connected layers for constructing high-level representation individually. These high-level representations are then fused using an LSTM

for optimal treatment or next visit prediction.
In addition, we introduce attention mechanism at the level of the variables. This allows to improve the model results and to make the model more interpretable. When using the attention at the level of the input variable, the model can identify the most important variables that are involved in the prediction. We can also deduce the importance of the different visits.

Given a set of patients $P = \{p_1, p_2, \ldots, p_r\}$, where r is the total number of patients.
Each patient p_x $(1 \leq x \leq r)$ is represented by a sequence of n visits v_t^x $(1 \leq t \leq n_x)$: $p_x = [v_1^x, \ldots, v_{n_x}^x]$.
The objective is to predict the optimal treatment among a set of treatments: $Y = \{y_1, \ldots, y_\varphi\}$, where φ is the number of treatments or the date of the next visit.

To deal with the prediction tasks, we have combined different information sources. Indeed, the availability different and complementary information sources can improve the prediction results.

Each source is built based on its impact on the disease to be represented differently.

We use the following three feature sources [1]:

- Personal features $(s(1))$: include gender, age, height and weight at the time of the visit.
- Clinical features $(s(2))$: include all clinical codes related to the diagnoses, the procedures of medications and the treatments;
- Comorbidity features $(s(3))$: refers to the presence of one or more disorders that occur at the same time as a primary disease.

To learn different representations from the sources, each information source is linked to fully connected layers where the neurons from one layer are connected to all the neurons of the next layer.

5.2.1 Information source representation with attention mechanism

Each visit v_t^x of a patient p_x at time t is composed of values of the three information sources: $v_t^x = (s(1)_t^x, s(2)_t^x, s(3)_t^x)$, where $s(1)_t^x$, $s(2)_t^x$ and $s(3)_t^x$ are respectively the personal, the clinical and the comorbidity feature vectors that represent the visit v_t^x of the patient p_x at time t.

Each features vector $s(k)_t^x$ $(k = \{1, 2, 3\})$ is represented as a vector of measurements: $s(k)_t^x = (s(k)_{t1}^x, s(k)_{t2}^x, \ldots, s(k)_{td_0^k}^x)$, where d_0^k is the number of features that represent the source $s(k)$.

We introduce a set of weights W^{α_k} $(k \in \{1, 2, 3\})$ that are learned to compute the variable-level attention. $\alpha(1)_t^x$, $\alpha(2)_t^x$ and $\alpha(3)_t^x$ are the attention vectors associated to the sources $s(1)_t^x$, $s(2)_t^x$ and $s(3)_t^x$ respectively. Each vector $\alpha(k)_t^x$ $(k \in \{1, 2, 3\})$ has the same dimension as its corresponding source since it represents the features associated to the source.

$$\alpha(k)_t^x = softmax(W^{\alpha_k} \cdot s(k)_t^x) \tag{5.1}$$

The features that constitute the patient visit will be weighted using the learned attention before the high-level representation learning.

$$\hat{s}(k)_t^x = \alpha(k)_t^x \odot s(k)_t^x \tag{5.2}$$

Where $\odot$ is an element-wise product. We obtain a high-level representation of the new source representation $\hat{s}(k)_t^x$ $(k \in \{1, 2, 3\})$ at time t using l_k fully connected hidden layers. Each hidden

[1] Our architecture can obviously integrate other information sources about a patient.

layer j is constituted of d_j^k neurons. For each source k, the activation of the i^{th} neuron $u(k)_{ti}^j$ from the layer j at time t is given by:

$$a(k)_{ti}^j = g(\sum_{z=1}^{d_{j-1}^k} w(k)_{zi}^j * a(k)_{tz}^{(j-1)} + b(k)_{ti}^j) \quad (1 \leq i \leq d_j^k) \tag{5.3}$$

where the $w(k)_{zi}^j$ is the weight of the connection from the neuron $u(k)_{tz}^{j-1}$ in the layer $(j-1)$ to the neuron $u(k)_{ti}^j$, $b(k)_{ti}^j$ is the bias associated to the neuron $u(k)_{ti}^j$, g is a nonlinear activation function.

For a patient p_x, the activation of the input layer of a source k at time t is given by: $a(k)_{ti}^0 = \hat{s}(k)_{ti}^x$.
[2]

The obtained high-level representations of the information contained in the visit v_t^x of a patient p_x at time t are concatenated to form one multidimensional vector $\tilde{v}_t^x$ that is defined as follows:

$$\tilde{v}_t^x = (a(1)_{t1}^{l_1}, \ldots, a(1)_{td_{l_1}^1}^{l_1}, a(2)_{t1}^{l_2}, \ldots, a(2)_{td_{l_2}^2}^{l_2}, a(3)_{t1}^{l_3}, \ldots, a(3)_{td_{l_3}^3}^{l_3}) \tag{5.4}$$

For the rest of the paper the components of $\tilde{v}_t^x$ are denoted $\tilde{v}_{tj}^x$, i.e.

$$\tilde{v}_t^x = (\tilde{v}_{t1}^x, \ldots, \tilde{v}_{td}^x), \quad where \quad d = d_{l_1}^1 + d_{l_2}^2 + d_{l_3}^3 \tag{5.5}$$

5.2.2 Temporal representation of the visits

For each patient, the high-level visit representations are processed sequentially by the LSTM network, as illustrated in Figure 5.1. Since we need a prediction at each time step, we use a One-to-One LSTM configuration. This means that for each input, the network associates an output.

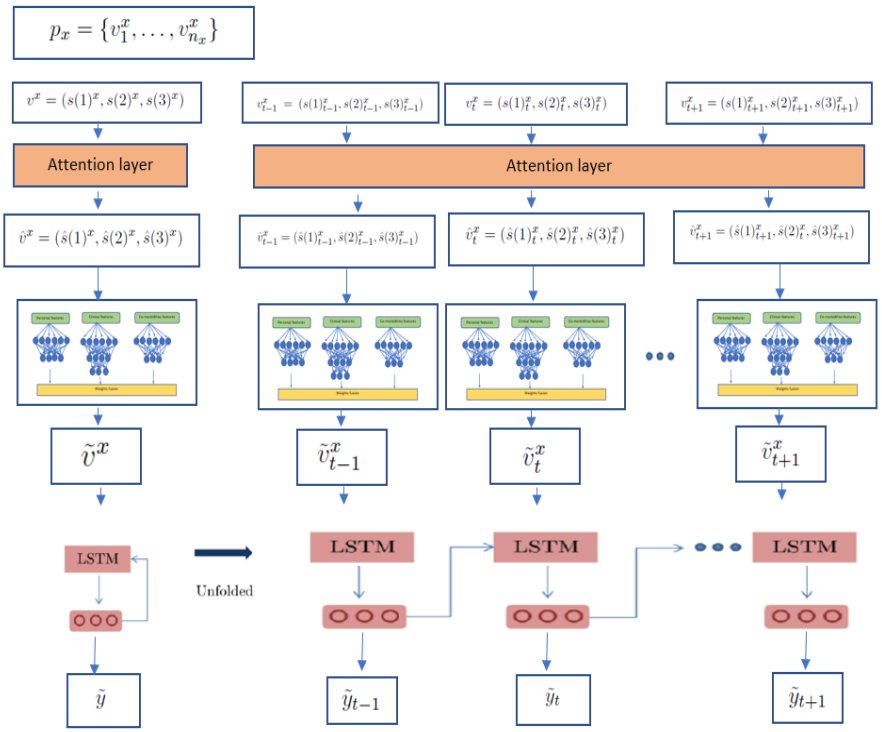

Figure 5.1: MS-LSTM architecture

[2]Note that the temporal aspect of the visits is not taken into account in the fully connected layers.

At time t, each visit v_t^x of patient $p_x \in P$ is transformed to a high-level representation vector $\tilde{v}_t^x$. This vector is embedded into an LSTM which outputs y_t^x that represents the optimal treatment or the date of the next visit of the patient.

As represented in Figure 5.2, an LSTM memory block consists of a memory cell c and three multiplicative gates which regulate the state of the cell: forget gate f, input gate i and output gate o. The memory cell encodes the knowledge of the inputs that have been observed up to the current time step. The forget gate controls whether the old information should be retained or forgotten. The input gate regulates whether new information should be added to the cell state while the output gate controls which parts of the new cell state to output.

At time t, the network takes three inputs: $\tilde{v}_t^x$, h_{t-1} and c_{t-1}. h_{t-1} is the state of the LSTM

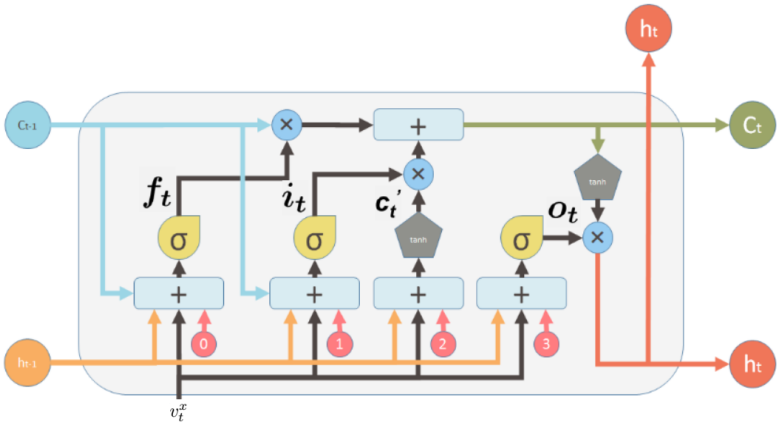

Figure 5.2: LSTM Unit

unit at time $(t-1)$ (the initial state $h_0 = 0$) and c_{t-1} is the memory cell representation at time $(t-1)$. h_t, c_t are respectively the state and the memory of the unit at the current time t. f_t, i_t and o_t that refer respectively to the forget, input and output gate are defined as follows:

$$f_t = \sigma(W_f[\tilde{v}_t^x; h_{t-1}] + b_f) \tag{5.6}$$

$$i_t = \sigma(W_i[\tilde{v}_t^x; h_{t-1}] + b_i) \tag{5.7}$$

$$o_t = \sigma(W_o[\tilde{v}_t^x; h_{t-1}] + b_o) \tag{5.8}$$

Where σ is the sigmoid activation function, W_t, W_i and W_o are the weights associated to the corresponding gates and b_f, b_i and b_o are the biases of the gates.

The current unit state h_t depends on the input $\tilde{v}_j^i$, the gate states and the previous state h_{t-1}. It is computed as follows:

$$h_t = o_t \cdot tanh(c_t) \tag{5.9}$$

Where $tanh$ is the hyperbolic tangent function, c_t is the current memory state, b_c its bias and $\tilde{c}_t$ the intermediate memory state. c_t and $\tilde{c}_t$ are calculated as follows:

$$\tilde{c}_t = tanh(W_c[\tilde{v}_t^x; h_{t-1}] + b_c) \tag{5.10}$$

$$c_t = f_t \cdot c_{t-1} + i_t \cdot \tilde{c}_t \tag{5.11}$$

The LSTM unit updates its state h_t at every time step t and propagates it to following output layer for the prediction of the optimal treatment or the date of the next visit.

The softmax output layer is used for a classification task. The output from the LSTM layer h_t is fed into the output layer to get the prediction probabilities:

$$\pi_t = sotmax(W.h_t + b) \tag{5.12}$$

Where π_t is the vector of predicted probabilities of the treatment prediction, W is the weight vector to be learned, and the b is the bias term.

The output of our approach is given by $\tilde{y}_t = argmax_{i=1}^{\varphi} \pi_{ti}$

To train the model, we use a categorical cross entropy function[3]

$$E = -\frac{1}{r}\sum_{x=1}^{r}\frac{1}{n_x}\sum_{t=1}^{n_x}\sum_{i=1}^{\varphi}[y_{ti}log(\pi_{ti})] \tag{5.13}$$

Where π_t is the model output probabilities and y_t, the true targets.

5.3 Experimental results

In this section, we present the experimental results which allow to validate our approach.
In the experiments we use the real dataset provided by CGEDIM (described in Chapter 4 section 4.4.2). This dataset describes the characteristics of 429,087 patients, each patient is represented by a series of visits. Each visit contains about 15 characteristics (see table 5.1).

5.3.1 Experiment Setup

The model was trained to predict the optimal treatment and the date of the next visit. The proposed model accepts each visit as a 15 dimensional vector v_t^x (15 features) which refers to the visit at time (t) of the patient p_x. As said before, a visit v_t^x is represented via Personal-derived features (s1), Clinical-derived features (s2) and Comorbidity-derived features (s3). Table 5.1 shows the features of each source.

First, we split the data into training, validation, and test sets. 60% patients were used for

Personal-derived features (s1)	Clinical-derived features (s2)	Comorbidity-derived features (s3)
Gender, Age_presc, Weight, Height	Box, Quantity, Frequency_label, Duration, Pulse, Diastolic pressure, Systolic pressure, Prescription_blood_sugar, last_measure_blood_sugar	Insulines_treatment, Other_treatment

Table 5.1: Sources features

training all models, 20% as the validation set and 20% as the test set. Then, we evaluated the final performances against the test set.
Each data source passed through an attention mechanism. The output of the attention mechanism is given as input to a two fully connected layers to be preprocessed: Layers 1 and 2 are fully connected dense layers and sigmoid activation functions. Finally, each visit of given patient is represented by appending the sources high-level representations. From this new representation, we make the predictions using an LSTM.
We trained our model for 1000 epochs (1000 iterations over the entire training data). To avoid overfitting, we used dropout between LSTM layers.
In order to validate our algorithm, we compared the results given by MS-LSTM with those that we obtained from:

[3]We can easily perform regression tasks by using linear activation function and mean squared error loss function.

(a) Doctor AI;

(b) Med2vect;

(c) Deepr;

(d) DeepCare;

(e) DeepPatient.

5.3.2 Experimental results

In order to validate our deep model, we evaluate the following aspects:

- Prediction results;
- Impact of pretreating data on the prediction results;
- Impact of source type on the prediction results;
- Identifying relevant clinical factors for test patients;
- Identifying an individual relevant clinical factors of a patient.

Prediction results

We have compared the results of different algorithms with MS-LSTM based attention. Since we are interested in predicting the optimal treatment and the date of the next visit, we reported the results in two tables.

- Predicting optimal treatment (T) (see table 5.2);
- Predicting the date of the next visit (V) (see table 5.3).

Approach	Accuracy	Precision	Recall	F1-Measure
MS-LSTM	**80.4** ±0.03	**78.4** ±0.02	**83.9** ±0.2	**81.0**±0.04
Doctor AI	**68.3** ±0.07	69.5±0.01	65.2±0.02	67.3±0.03
M2vect	65.4 ±0.02	59.5±0.07	69.2±0.11	64.0±0.05
DeepCare	55.5±0.11	54.6±0.03	64.9±0.01	59.3±0.09
Deepr	51.9 ±0.04	52.2±0.02	46.6±0.03	49.2±0.02
DeepPatients	58.8 ±0.01	57.2±0.03	69.3±0.01	62.7±0.13

Table 5.2: Comparison of algorithms when predicting optimal treatment

Table 5.2 confirms that the proposed approach is able to outperform the existing ones by a large margin while predicting optimal treatment. MS-LSTM obtains an accuracy of up to 80.4% while Doctor AI is lower than 68.3%. This can be explained by using LSTM against GRU in DoctorAI. In fact, in theory LSTMs remember longer sequences than GRUs and outperform them in tasks requiring modeling long-distance relations. In addition, the introduction of attention mechanism allows to improve the results of LSTM models.
The results were in our favor for predictive accuracy, comparing with other approaches, which confirms that our visit representation with an attention mechanism and a fully connected layers is efficient since it improves the predictions.
Predicting the date of the next visit should be modeled as a regression problem. The duration between visits can be highly skewed since it depends on the availability of doctors and patients, in addition to the patient health state. In our approach, we decided to discretize the duration between visits. Thus, the regression problem is transformed to classification problem.
After testing several discretization possibilities, we opted for the discretization by month. Since in our EHR data minimum duration between visits is equal to 10 days and the maximum between visits is up to 200, we defined 7 classes:

- Class 1: duration $\leq$ 1 month;
- Class 2: 1 month $<$ duration $\leq$ 2 months;
- Class 3: 2 months $<$ duration $\leq$ 3 months;
- Class 4: 3 months $<$ duration $\leq$ 4 months;
- Class 5: 4 months $<$ duration $\leq$ 5 months;
- class 6: 5 months $<$ duration $\leq$ 6 months;
- Class 7: duration $>$ 6 months.

Approach	Accuracy	Precision	Recall	F1-Measure
MS-LSTM	**79.5** ±0.04	**78.6** ±0.01	**80.9** ±0.01	**79.7**±0.03
Doctor AI	68.2 ±0.02	66.7 ±0.11	72.2 ±0.04	69.4 ±0.03
M2vect	57.6 ±0.01	56.2 ±0.01	66.8 ±0.1	61.1 ±0.03
DeepCare	74.7 ±0.02	73.4 ±0.01	77.1 ±0.05	75.2 ±0.01
Deepr	59.4 ±0.03	58.2 ±0.05	65.6 ±0.07	61.7 ±0.01
DeepPatients	62.2 ±0.02	60.3 ±0.03	70.9 ±0.02	65.2 ±0.01

Table 5.3: Comparison of algorithms when predicting the duration till the next visit.

Table 5.3 illustrates the superiority of the hierarchical architecture of MS-LSTM algorithm since it gives an accuracy up to 78.6% which is higher than those obtained by Doctor AI (accuracy = 68.2%), M2vect (accuracy = 68.2%), Deepr (accuracy = 59.4%), DeepPatients (accuracy = 62.2%) and DeepCare (accuracy up to 74.7%).

Impact of source type on the prediction results

To test whether combining multiple sources increases performance, we have tested all possible combinations (which would require 7 experiments, given 3 different input sources). In fact, identifying the significant sources of features plays an important role in predicting hypertension disease. It is crucial to select the correct combination of significant features which can improve the performance of our prediction model. Table 5.4 shows that Clinical features outperform the personal and the comorbidity features whether when predicting the optimal treatment or the date of the next visit. They improve prediction accuracy by up to 44.7% while including them when predicting the optimal treatment. However, using only personal features gives only 24.5% of accuracy. This can be explained by the fact that age, gender, weight etc. without blood pressure measurements do not really make sense for treatment prediction. Finally, the results show that we can produce more accurate predictions when merging all information sources since they provide measurements that can be different and complementary in their nature (accuracy=80.4% for the prediction of the optimal treatment).
Including only Clinical data in the prediction of the duration till the next visit gives 51.8% of accuracy which is much higher that the accuracy of only including personal data(19.4%)or the accuracy of just including comorbidity features (13.7%).

Features (1:included, 0:not included)			Accuracy (T)	Accuracy (V)
Personal	Clinical	Cormobidities		
0	0	1	22.2 ±0.03	13.7±0.04
0	1	0	49.3 ±0.01	51.8±0.03
0	1	1	64.3±0.02	59.1±0.05
1	0	0	24.5±0.04	19.4±0.03
1	0	1	36.3±0.06	46.2 ±0.02
1	1	0	67.7 ±0.07	57.1±0.01
1	1	1	**80.4** ±0.03	**79.5** ±0.04

Table 5.4: Impact of source type on the prediction results

Impact of pretreating data on the prediction results

In order to show the impact of preprocessing the data sources, we tested our model with and without pretreating features with fully connected layers. Table 5.5 shows that pretreating features from EHR, provides new data representation and improves prediction results whether when predicting the optimal treatment or the date of the next visit. This can be explained by the ability of our approach to build a high-level representation from the initial noisy input data sources using the fully connected layers.

cases	Accuracy (T)	Accuracy (V)
Without pretreating features	69.3 $\pm$0.02	65.7$\pm$0.03
With pretreating features	**80.4**$\pm$0.03	**79.5** $\pm$0.04

Table 5.5: Impact of pretreating data on the prediction results

Identifying relevant clinical factors

The attention scores over visits were used to construct heatmaps for the models trained to each of the two prediction problems. The softmax scores for each feature are averaged over all test patients to obtain patients-averaged attention maps demonstrating when individual predictor variables had the most influence on each prediction. Figure 5.3 (A) shows the produced heatmap by averaging the attention scores of all test patients when predicting the optimal treatment at each visit. This shows, for each visit, the most relevant features that lead to the prediction. (B) shows the averaged attention scores obtained in (A) on the features for each patient visit. This reveals the visits which contributed the most for the prediction. We can see that Visits 4 and 10 are the most efficient visits for the prediction of optimal treatment. (C) shows the averaged attention scores on the visits to highlight the most relevant features that are used for the prediction. As we can see, the most significant features are 'Pulse' with an attention score equal to 0.63, 'Diastolic pressure' with an attention score equal to 0.62. The feature 'Systolic pressure' gives an attention score up to 0.58. Overall, we can notice that the most relevant attributes are clinical features.
Lastly, the attention mechanism can be also used to observe the importance of sources. By averaging the scores of the features that constitute each data source, we obtained their importance scores. As we can see, (D) presents the most important source for prediction of the optimal treatment is the clinical features with an attention score up to 0.53.
The second heatmap illustrated in Figure 5.4 shows the patients-averaged attention scores when predicting the date of the next visit. As explained previously, we can extract features with the highest attention scores for each visit from (A), the most relevant visits from (B), the most important features for the prediction from (C) and the important data sources from (C). As we can see in (B), the visits 9, 10 and 11 contribute the most to this prediction task. (C) shows that the features 'Quantity', 'Diastolic pressure' and 'Pulse' are the most relevant features that lead to the prediction of the next visit with respectively $0,616$, $0,61$ and $0,60$ as attention scores.
Similarly, the most important source for this prediction is the clinical features source with an attention score up to 0.54.

Identifying relevant clinical factors for individual patient

Individualized predictions and attention visualization maps can be generated for each patient for each prediction task. Figures 5.5 and 5.6 show the same information as previousely for each patient. Instead of averaging over all test patients, in (A) we built heatmaps for each patient individually. We thus ploted the attention scores related to visits (B), features (C) and data sources (D). Figure 5.5 (B) shows that visit 5 is the most relevant one for the prediction of the optimal treatment for this patient. (C) shows that 'Diastolic pressure', 'Pulse' and 'Prescription_blood_sugar' are the most important features with respectively 0.78, 0.74 and 0.74 as

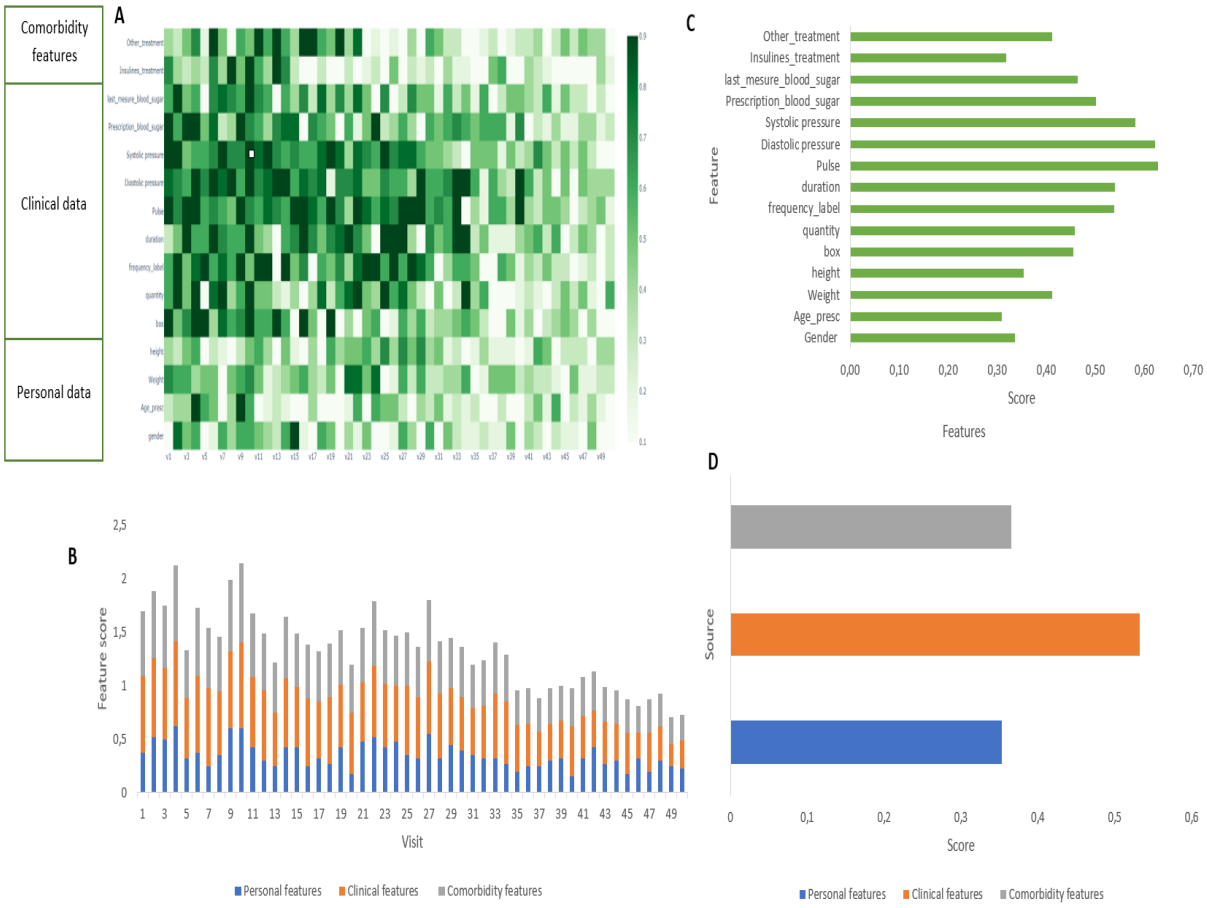

Figure 5.3: Averaged Attention scores over all test patients when predicting the optimal treatment: (A) shows the produced heatmap by averaging the attention scores of all test patients. (B) shows the averaged of attention scores on the features for each patients visit. (C) shows the averaged attention scores on the visits to highlight the most relevant features. (D) presents the most important sources for the prediction.

attention scores. (D) confirms that the most important features belong to the clinical features with an attention score up to 0.7.

Similarly, heatmap illustrated in Figure 5.6 shows that for the same patient, we found that visit 4 is the most important visit (Figure 5.6 (B)), the features 'Duration' and 'Diastolic pressure' are the most important features for the prediction of the date of the next visit with respectively 0.71 and 0.60 as attention scores ((Figure 5.6 (C)). Finally, Figure 5.6 (C) illustrates the sources scores and confirms that clinical features represent the most important data source with an attention score up to 0.47.

5.3.3 Discussion

One of the main challenges in using deep algorithms with EHRs is scalability. The largest datasets used in the literature are those used in Ranganath et al [221] with 13,180 patients and in Choi and al. [60] with 263,706 patients. We can see that our dataset, described in chapter 4, section 4.4.2 is larger than those datasets and thus expect than the comparison results are more confident. Another limitation includes building a predictive model by using either only demographic features [26] or clinical features [98]. The former discards a huge proportion of information in each patient record, while the latter ignores knowledge and guidelines coming from human intelligence. We use comorbidity information to improve the performance of our algorithm. Such expert knowledge is considered as a domain-specific knowledge which is also a big challenge since it impacts the progression of disease and patient state.

When working with DL models in predicting clinical events with EHRs, it is important to ensure that the model is flexible. That means the possibility of adding new data sources. In fact,

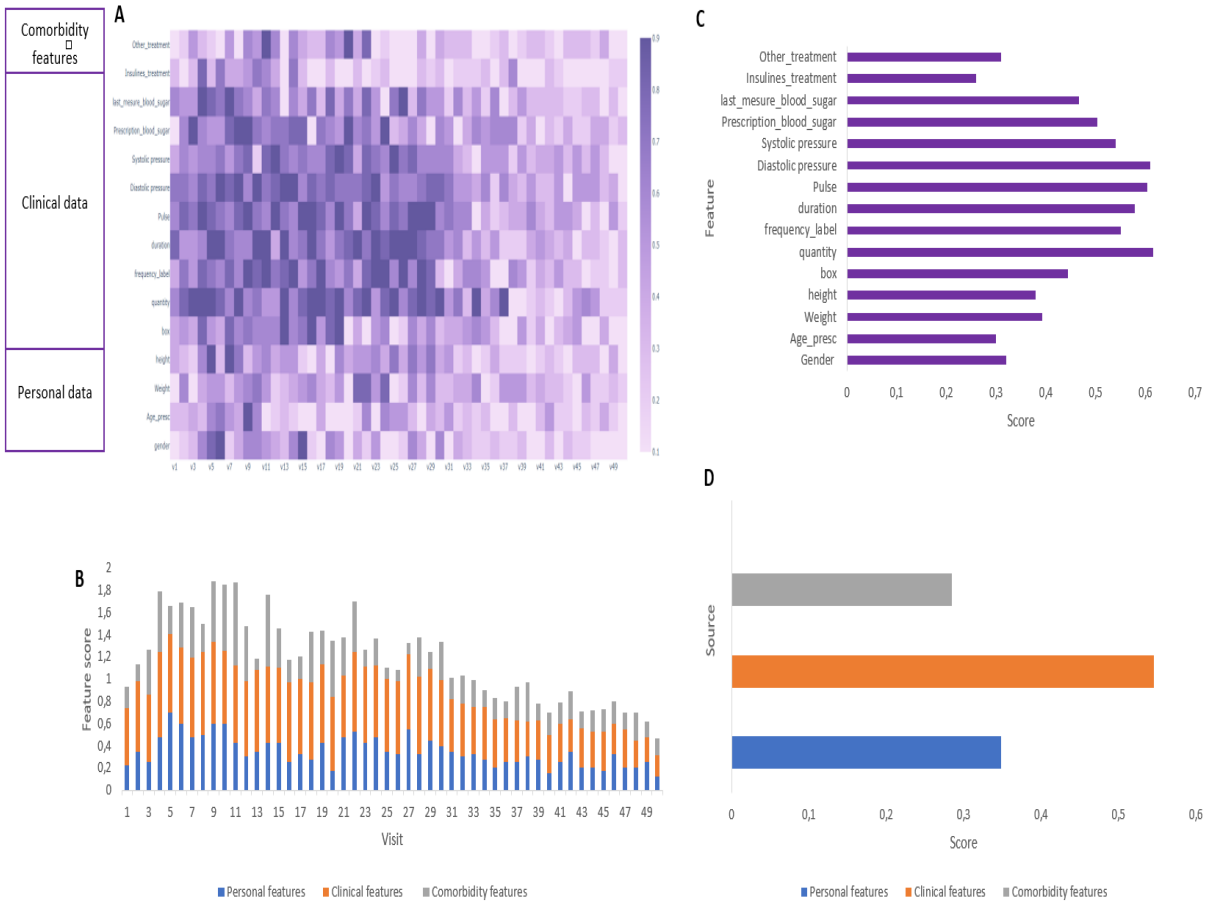

Figure 5.4: Averaged Attention scores over all test patients when predicting the date of the next visit: (A) shows the produced heatmap by averaging the attention scores of all test patients. (B) shows the averaged of attention scores on the features for each patients visit. (C) shows the averaged attention scores on the visits to highlight the most relevant features. (D) presents the most important sources for the prediction.

the combination of multiple data: biological, pathological, their evolution, raises expectations and hopes in terms of understanding the causes and mechanisms of diseases as well as for the personalization of medical monitoring. Some of existing works did not take into account the integration of new data sources such as the work in [192]. MS-LSTM can easily add new data source even if it is heterogeneous, discrete, or categorical since it pretreats each data source differently and separately to adapt them to visit representation at the time of prediction.
This study presents a deep learning model based on LSTM and attention mechanism to predict optimal treatment and the date of the next visit. The model achieves high ability with an accuracy up to 80.4% when predicting the optimal treatment and an accuracy up to 72.3% when predicting the date of the next visit. We showed that pretreating personal, clinical and comorbidity features from EHR using a fully connected layers, with an an attention mechanism outperforms models that ignore any of these characteristics. Also, we showed that combining different features results gives better performance than using either of them alone. In addition, our model can capture the significant features, sources or visits that contribute the most for the prediction due to the attention mechanism. Thus, our model can easily provide an interpretable prediction.

5.4 Conclusion

In this chapter, we have proposed MS-LSTM model, which uses a LSTM and an attention mechanism to predict the optimal treatment and the date of the next visit. We have tested MS-LSTM on a large real EHR dataset, it significantly outperformed many existing approaches. We have also shown that combining different sources of variables with an attention mechanism

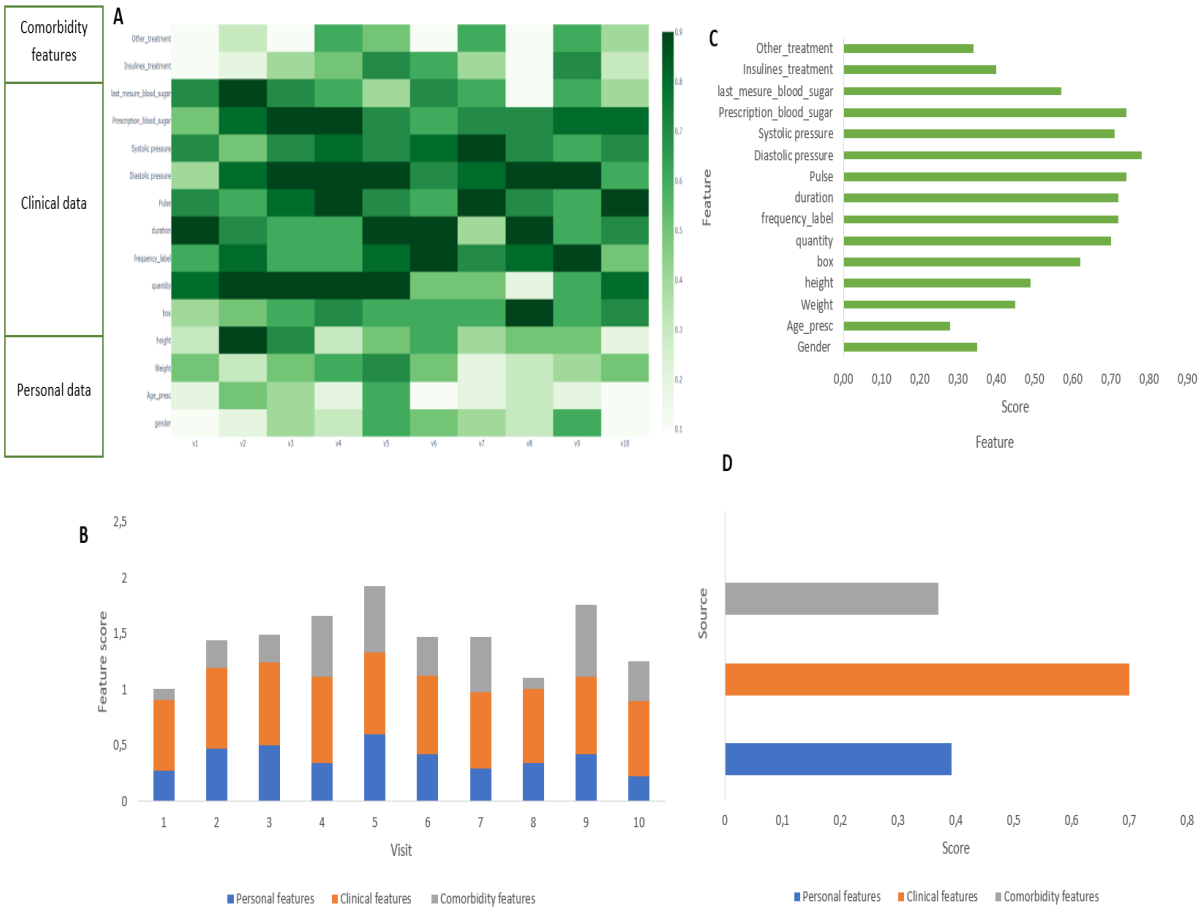

Figure 5.5: Personalized attention scores for one patient with 10 visits for the prediction of the optimal treatment: (A) shows the produced heatmap for attention scores over 10 visits. (B) shows the averaged attention scores on the features for each visit. (C) shows the averaged attention scores on the visits to highlight the most relevant features. (D) presents the most important sources for the prediction.

and a LSTM network highly improve the performance. In addition, MS-LSTM can select the relevant factors that contribute to the prediction which is important in the medical field.

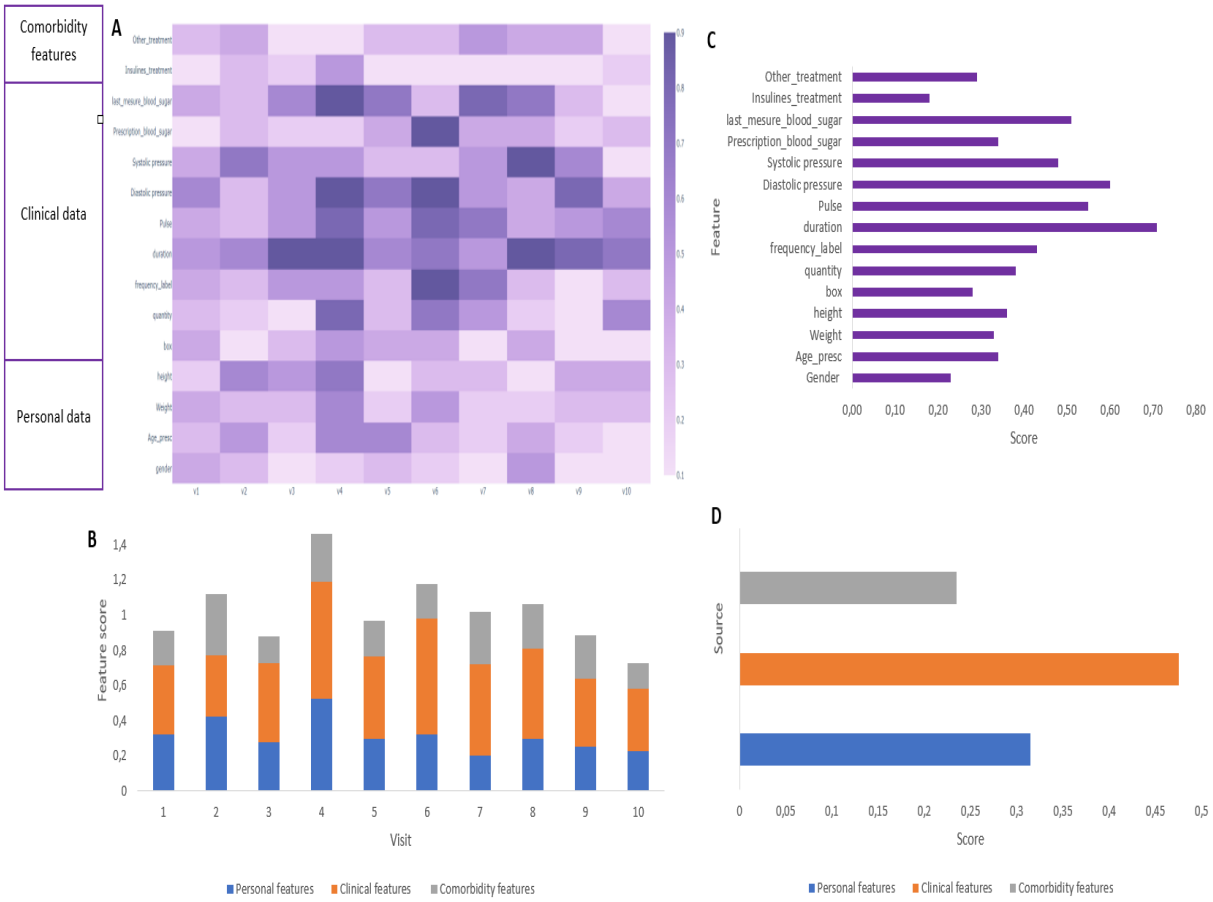

Figure 5.6: Personalized attention scores for one patient with 10 visits for the prediction of the date of the next visit: (A) shows the produced heatmap for attention scores over 10 visits. (B) shows the averaged attention scores on the features for each visit. (C) shows the averaged attention scores on the visits to highlight the most relevant features. (D) presents the most important sources for the prediction.

Chapter 6

Conclusion and prospects

Contents

6.1 Conclusion

The first contribution in this thesis is an original approach to design a medical support system for the individualized follow-up of patients with hypertension. From data collected on several patients, the objective was to define patient profiles and to find the optimal treatment and to predict the date of the next visit. Based on ensemble methods which improve the results of learning algorithms, we were first interested in combining deep neural networks. Traditional ensemble methods are based on a simple majority vote or a simple average of the classifiers predictions, which makes difficult to explain the result. In addition, these methods only exploit the results of the classification or regression, rather than the internal knowledge learned by the ML models and do not allow prior knowledge injection.
Using multiagent argumentation, we exploited classification knowledge as argument for negotiation between agents. Indeed, the idea of combining several DMPLs is not new, but the association of a DMLP to an agent and more precisely the use of argumentation between "DMLP agents" introduced intelligence into the way of combining DMLPs. The experimental results show an improvement of the prediction results since for each prediction, it is the DMLP that have the most robust argument that is used. In addition, the user is able to judge the acceptability or not of a result since the latter has an explanation (argument).
Moreover, the originality of our approach lies in the fact of automatically constructing the negotiation arguments from training data. These arguments are then used by an expert system, itself embedded in an agent, to trigger reasoning on the arguments. Moreover, other agents containing rules about domain knowledge can be easily added.
DL models are obtained from training large amount of data. This purely data-driven learning may induce contradictory results that can be uninterpretable. Our medical support system can overcome this weakness by injecting prior knowledge to guide the learning step of models and reduce their uninterpretability. Thus, the expert can control the decision-making system either by injecting prior knowledge or eliminating conflicting knowledge.
In order to validate our system, we used different public dataset, virtual data and real EHR data provided by CGEDIM (medical prescription platform used by 23000 doctors in France), which has been collected from 3000 doctors. This dataset describes the characteristics of 429087 patients with hypertension, each patient is represented by a series of visits. Experiments show that, as ensemble method, our approach significantly outperforms single classifiers and traditional ensemble methods. In addition, our method effectively provides explanation behind decisions and therefore addresses the recent need for Explainable AI. The explanation provided to the user is easy to grasp so he/she is able to judge the acceptance of decisions.
The "learning" phase that allows to build knowledge bases, is very important in our system. In this phase, the choice of the ML model, which represents the basic building block of our architecture is crucial. In the actual version of our system, we used several DMLPs for learning non temporal data which provides significant results. However, DMLP is not suitable for time series since it ignores an important piece of information in EHR embedded in the temporal trajectory of patient [64], [165]. EHRs include a sequence of measurements (clinical visits) over the time which contains important information about the progression of disease and patient state. Capturing the visits sequentially (as they appear in EHR) adds a significant precision compared to a memory-less neural network like multilayer perceptron. Thus, it was necessary to build another model able to consider the temporal trajectory of EHR.
For this purpose, we designed an original Recurrent neural network model called MS-LSTM based attention which was our second contribution. Due to the power of LSTM, MS-LSTM can consider the sequential trajectory of the visits which adds a significant precision compared to a memory-less neural network. In fact, this sequence of measurements over time contains important information about the progression of disease. We have also addressed the issue combining different information sources to deal with the prediction task and more particularly in the field of personalized medicine. Indeed, it is admitted that the heterogeneity of the medical information (personal information, measurements, diagnosis, comorbidities, etc.) can improve predictions.
The proposed model can predict the optimal treatment and the date of the next visit. An attention mechanism was applied to the input data to capture the most relevant features. Different data sources are pretreated differently and merged to form the visit representation vector. We

have combined several information sources because they provide measurements that can be different and complementary in their nature. It is therefore crucial considering the integration issue during the conception of prediction method since, the way in which different data is combined can drastically impact the result. Then the resulting visit representation vectors are fed one by one into an LSTM. The output of the model can be the optimal treatment or the duration till the next visit.
Our new architecture-based LSTM has shown a great and meaningful results prediction using the EHR data described above. Moreover, MS-LSTM can easily provide what variable contributed the most to the prediction.

6.2 Prospects

The works achieved during this thesis are a good starting point towards the prediction of clinical events. These works overcome some limitations of the state of the art. In this section, we will discuss about some possible improvements.
Our Medical support system deals with an argumentative view of decision making, thus focusing on the issue of justification the best decision made in each situation. Such an approach has indeed some obvious benefits. On the one hand, not only the best choice is suggested to the user, but also the reasons of this recommendation can be provided in an easy-to-understand format. On the other hand, our decision making system allows injecting recommended knowledge given by a domain expert. Some possible improvements could be done in the future:

(a) New data
Currently, we have validated our approach using EHR Gers Data provided by CGEDIM and collected from 3000 doctors in France. In the short term, it will be a question of carrying out experiments on a large scale of data from self-measurements of voluntary patients to consolidate the results of our algorithms and our approach. The comparison of the results of the two datasets could be an interesting study about the progression of the disease and the quality of measurements.

(b) Applying other learning algorithms
We have already use DMLP and LSTM for the prediction of medical events related to the hypertension. In the next step, we plan to integrate other learning algorithms which are able to treat and model temporal data. The choice of which model to use depends on several factors such as the data used to build and train these models and the choice of prediction objectives. For example, we used LSTMs to model the temporal trajectory of patient visits. However, the sequential information can also be modelled by other models such as GRUs or CNNs, etc. We plan to use other neural networks and provide other variants of our architecture. Then, since the parameterization of the learning algorithms is crucial, it will be a question of using heuristics to improve the choice of these parameters.

(c) Improvement of the argumentation process
Agent negotiation is a current research topic. In this thesis the argumentation process is limited to an exchange of arguments between agents based on the speech acts. More complex description logics which will allow for example the revision of the beliefs can be used. The use of production rules and an expert system can be generalized to improve the negotiation between agents. We can also use coalitions in communication between agents or another more complex communication protocol. The formation of coalitions [248] is another approach in terms of the functioning of interactions between agents and collective problem solving. For a group of agents faced with a request, this involves making individual compromises to reach a consensus that is satisfactory for all parties (ideal case). For example, if two agents must choose a color and one prefer black, the other for white, the consensus and the final choice of the two agents may be gray. If more than two agents are involved, alliance mechanisms can be introduced to reach consensus more quickly. The difficulty will then be to define the communication protocol adequately [243]. The protocol must both allow agents to share their current choices with each other and modify those

choices until consensus is reached. There is no ideal solution as there are so many decisions about how the system will organize itself. These decisions are ultimately the responsibility of the modeler.

(d) Validation diagnosis by a medical expert
The system provides the decisions (diagnosis) with explanation. Those decisions need to be validated by the doctor. Explanation will give him the information to confirm or refute the diagnosis. And therefore, the combination between the capacity of the model and the predictive power of doctors, will provide diagnosis results that are clinically meaningful. It will be intresting also to validate our approach by including a theoretical analysis in order to verify for example the coherence of rules.

(e) Handling missing data
Missing data is a common problem with time series data. Currently, input data with missing values are removed from the datasets. But this way to do is not optimal. In the articles [200] [237] [120], the authors present several approaches to handle missing data. In future work, we plan to study these approaches and choose the most efficient one.

(f) Integration of new data sources
In this work, we tested different data sources such as personal, clinical and the comorbidity data. These data sources will not be sufficient for other applications like adapting diagnosis to the preferences of the patients. Indeed, it is recommended that hygienic and dietary measures be initiated depending on the patient's profile, the severity of his hypertension, his preferences, his adherence to these measures, the time taken to initiate drug treatment and may be his genomic data will be adapted to reach the goal of controlled hypertension.

(g) Knowledge Extraction from MS-LSTM
Extracting understandable knowledge from LSTM solves two fundamental problems: it provides insight into the logic of the network and in many cases, it improves the capacity of the network to generalize the knowledge gained. The extracted knowledge implicitly encoded in LSTM can take the form of an automata or rules [144]. Rules are very general structures that provide an easy to understand prediction. Extracted knowledge from LSTM will make possible the integration of the model into our decision support system described in chapter 4.

(h) Knowledge prior injection
In this work, prior knowledge was injected into an agent to guide the argumentation process in the medical support system. Next step will be may be injecting learned knowledge from one doctor experience or prior knowledge from official guidelines into MS-LSTM itself to guide the learning step which is guided by data.

Bibliography

[1] Nadia Abchiche-Mimouni, Antonio Andriatrimoson, Etienne Colle, and Simon Galerne. Coalaa-gen: A general adaptive approach for ambient assistive applications. In KES, pages 324–334, 2016.

[2] Rakesh Agrawal, Tomasz Imieliński, and Arun Swami. Mining association rules between sets of items in large databases. SIGMOD Rec., 22(2):207–216, June 1993.

[3] Malik Al Qassas, Daniela Fogli, Massimiliano Giacomin, and Giovanni Guida. Analysis of clinical discussions based on argumentation schemes. Procedia Computer Science, 64:282–289, 2015.

[4] Malik Al Qassas, Daniela Fogli, Massimiliano Giacomin, and Giovanni Guida. Argmed: A support system for medical decision making based on the analysis of clinical discussions. In Real-World Decision Support Systems, pages 15–41. Springer, 2016.

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

[6] L Amgoud and C Cayrol. On the acceptability of arguments in preference-based argumentation. inproceedings of the 14th annual conference on uncertainty in artificial intelligence (uai-98), san francisco, ca, usa, 1998.

[7] Leila Amgoud. Contribution a l'integration des préferences dans le raisonnement argumentatif. PhD thesis, PhD thesis, Université Paul Sabatier, Toulouse, 1999.

[8] Leila Amgoud and Claudette Cayrol. Integrating preference orderings into argument-based reasoning. In Qualitative and Quantitative Practical Reasoning, pages 159–170. Springer, 1997.

[9] Leila Amgoud and Claudette Cayrol. Inferring from inconsistency in preference-based argumentation frameworks. Journal of Automated Reasoning, 29(2):125–169, 2002.

[10] Leila Amgoud and Claudette Cayrol. A reasoning model based on the production of acceptable arguments. Annals of Mathematics and Artificial Intelligence, 34(1-3):197–215, 2002.

[11] Leila Amgoud, Simon Parsons, and Nicolas Maudet. Arguments, dialogue, and negotiation. In ECAI, 2000.

[12] Muhammad Amin and Amir Ali. Application of multilayer perceptron (mlp) for data mining in healthcare operations. In 3rd International Conference on Biotechnology, page 9, University of South Asia, Lahore, Pakistan, 02 2017.

[13] Rick Anderson. Rnn, talking about gated recurrent unit. 2019.

[14] Robert Andrews, Joachim Diederich, and Alan B. Tickle. Survey and critique of techniques for extracting rules from trained artificial neural networks. Knowledge-Based Systems, 8(6):373 – 389, 1995. Knowledge-based neural networks.

[15] Ofer Arieli and Martin Caminada. A general qbf-based formalization of abstract argumentation theory. Computational Models of Argument, (245):105–116, 2012.

[16] Awais Ashfaq, Anita Sant'Anna, Markus Lingman, and Sławomir Nowaczyk. Readmission prediction using deep learning on electronic health records. Journal of biomedical informatics, 97:103256, 2019.

[17] American Diabetes Association et al. Treatment of hypertension in adults with diabetes. Diabetes care, 26(suppl 1):s80–s82, 2003.

[18] Katie Atkinson, Trevor Bench-Capon, and Sanjay Modgil. Argumentation for decision support. In International Conference on Database and Expert Systems Applications, pages 822–831. Springer, 2006.

[19] M. Gethsiyal Augasta and T. Kathirvalavakumar. Reverse engineering the neural networks for rule extraction in classification problems. Neural Processing Letters, 35(2):131–150, Apr 2012.

[20] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. PloS one, 10(7):e0130140, 2015.

[21] Wenjia Bai, Ozan Oktay, Matthew Sinclair, Hideaki Suzuki, Martin Rajchl, Giacomo Tarroni, Ben Glocker, Andrew King, Paul M. Matthews, and Daniel Rueckert. Semi-supervised learning for network-based cardiac mr image segmentation. In Maxime Descoteaux, Lena Maier-Hein, Alfred Franz, Pierre Jannin, D. Louis Collins, and Simon Duchesne, editors, Medical Image Computing and Computer-Assisted Intervention MICCAI 2017, pages 253–260, Cham, 2017. Springer International Publishing.

[22] Trapit Bansal, David Belanger, and Andrew McCallum. Ask the gru: Multi-task learning for deep text recommendations. In Proceedings of the 10th ACM Conference on Recommender Systems, pages 107–114, 2016.

[23] G Octo Barnett, James J Cimino, Jon A Hupp, and Edward P Hoffer. Dxplain: an evolving diagnostic decision-support system. Jama, 258(1):67–74, 1987.

[24] Pietro Baroni, Martin Caminada, and Massimiliano Giacomin. An introduction to argumentation semantics. Knowledge Engineering Review, 26(4):365, 2011.

[25] Pietro Baroni and Massimiliano Giacomin. Solving semantic problems with odd-length cycles in argumentation. In European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty, pages 440–451. Springer, 2003.

[26] Senjuti Basu Roy, Ankur Teredesai, Kiyana Zolfaghar, Rui Liu, David Hazel, Stacey Newman, and Albert Marinez. Dynamic hierarchical classification for patient risk-of-readmission. In Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '15, page 1691–1700, New York, NY, USA, 2015. Association for Computing Machinery.

[27] Trevor JM Bench-Capon and Paul E Dunne. Argumentation in artificial intelligence. Artificial intelligence, 171(10-15):619–641, 2007.

[28] Emelia J Benjamin, Michael J Blaha, Stephanie E Chiuve, Mary Cushman, Sandeep R Das, Rajat Deo, Sarah D De Ferranti, James Floyd, Myriam Fornage, Cathleen Gillespie, et al. Heart disease and stroke statistics—2017 update. 2017.

[29] Eta S Berner. Clinical decision support systems: state of the art. AHRQ publication, 90069:1–26, 2009.

[30] Philippe Besnard and Anthony Hunter. A logic-based theory of deductive arguments. Artificial Intelligence, 128(1-2):203–235, 2001.

[31] Philippe Besnard and Anthony Hunter. Elements of argumentation, volume 47. MIT press Cambridge, 2008.

[32] Thulasi Bikku. Multi-layered deep learning perceptron approach for health risk prediction. Journal of Big Data, 7(1):1–14, 2020.

[33] Marc Billaud and Xavier Guchet. L'invention de la médecine personnalisée-entre mutations technologiques et utopie. médecine/sciences, 31(8-9):797–803, 2015.

[34] Elizabeth Black, Anthony Hunter, and Jeff Z Pan. An argument-based approach to using multiple ontologies. In International Conference on Scalable Uncertainty Management, pages 68–79. Springer, 2009.

[35] David Blumenthal and Marilyn Tavenner. The "meaningful use" regulation for electronic health records. New England Journal of Medicine, 363(6):501–504, 2010.

[36] Guillaume Bobrie, Pierre Clerson, Joel Menard, Nicolas Postel-Vinay, Gilles Chatellier, and Pierre-Francois Plouin. Masked hypertension: a systematic review. Journal of hypertension, 26(9):1715–1725, 2008.

[37] Guido Bologna and Yoichi Hayashi. A rule extraction study on a neural network trained by deep learning. In 2016 International Joint Conference on Neural Networks, IJCNN 2016, Vancouver, BC, Canada, July 24-29, 2016, pages 668–675. IEEE, 2016.

[38] Guido Bologna and Yoichi Hayashi. A comparison study on rule extraction from neural network ensembles, boosted shallow trees, and svms. 2018:1–20, 01 2018.

[39] Andrei Bondarenko, Phan Minh Dung, Robert A Kowalski, and Francesca Toni. An abstract, argumentation-theoretic approach to default reasoning. Artificial intelligence, 93(1-2):63–101, 1997.

[40] Elise Bonzon and Nicolas Maudet. On the outcomes of multiparty persuasion. In Proceedings of the 8th International Conference on Argumentation in Multi-Agent Systems, ArgMAS'11, pages 86–101, Berlin, Heidelberg, 2012. Springer-Verlag.

[41] Olcay Boz and Donald Hillman. Converting a trained neural network to a decision tree dectext-decision tree extractor. Citeseer, 2000.

[42] Ivan Bratko, Martin Možina, and Jure Žabkar. Argument-Based Machine Learning, pages 11–17. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.

[43] Lars Braubach, Alexander Pokahr, Daniel Moldt, and Winfried Lamersdorf. Goal representation for bdi agent systems. In International Workshop on Programming Multi-Agent Systems, pages 44–65. Springer, 2004.

[44] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. Classification and Regression Trees. Wadsworth and Brooks, Monterey, CA, 1984.

[45] Leo Breiman. Bagging predictors. Machine learning, 24(2):123–140, 1996.

[46] Leo Breiman and Leo Breiman. Bagging predictors. In Machine Learning, pages 123–140, 1996.

[47] Brian J. Buta, Jeremy D. Walston, Job G. Godino, Minsun Park, Rita R. Kalyani, Qian-Li Xue, Karen Bandeen-Roche, and Ravi Varadhan. Frailty assessment instruments: Systematic characterization of the uses and contexts of highly-cited instruments. Ageing Research Reviews, 26:53 – 61, 2016.

[48] Davide Calvaresi, Michael Schumacher, Mauro Marinoni, Roger Hilfiker, Aldo F Dragoni, and Giorgio Buttazzo. Agent-based systems for telerehabilitation: strengths, limitations and future challenges. In Agents and multi-agent systems for health care, pages 3–24. Springer, 2017.

[49] Martin Caminada. On the issue of reinstatement in argumentation. In European Workshop on Logics in Artificial Intelligence, pages 111–123. Springer, 2006.

[50] Martin Caminada. Semi-stable semantics. COMMA, 144:121–130, 2006.

[51] Robert J Carroll, Anne E Eyler, and Joshua C Denny. Naïve electronic health record phenotype identification for rheumatoid arthritis. In AMIA annual symposium proceedings, volume 2011, page 189. American Medical Informatics Association, 2011.

[52] Cristiano Castelfranchi. Commitments: From individual intentions to groups and organizations. In ICMAS, volume 95, pages 41–48, 1995.

[53] Claudette Cayrol. On the relation between argumentation and non-monotonic coherence-based entailment. In IJCAI, volume 95, pages 1443–1448, 1995.

[54] Claudette Cayrol and Marie-Christine Lagasquie-Schiex. Graduality in argumentation. Journal of Artificial Intelligence Research, 23:245–297, 2005.

[55] Claudette Cayrol and Marie-Christine Lagasquie-Schiex. Bipolar abstract argumentation systems. In Argumentation in Artificial Intelligence, pages 65–84. Springer, 2009.

[56] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining, pages 785–794, 2016.

[57] Yu Cheng, Fei Wang, Ping Zhang, and Jianying Hu. Risk prediction with electronic health records: A deep learning approach. In Proceedings of the 2016 SIAM International Conference on Data Mining, pages 432–440. SIAM, 2016.

[58] Carlos Iván Chesñevar, Ana Gabriela Maguitman, and Ronald Prescott Loui. Logical models of argument. ACM Computing Surveys (CSUR), 32(4):337–383, 2000.

[59] Arun Chockalingam, Norman R Campbell, and J George Fodor. Worldwide epidemic of hypertension. Canadian journal of cardiology, 22(7):553–555, 2006.

[60] Edward Choi, Mohammad Taha Bahadori, Andy Schuetz, Walter F. Stewart, and Jimeng Sun. Doctor ai: Predicting clinical events via recurrent neural networks. In Finale Doshi-Velez, Jim Fackler, David Kale, Byron Wallace, and Jenna Wiens, editors, Proceedings of the 1st Machine Learning for Healthcare Conference, volume 56 of Proceedings of Machine Learning Research, pages 301–318, Children's Hospital LA, Los Angeles, CA, USA, 18–19 Aug 2016. PMLR.

[61] Edward Choi, Mohammad Taha Bahadori, Elizabeth Searles, Catherine Judith Coffey, and Jimeng Sun. Multi-layer representation learning for medical concepts. ArXiv, abs/1602.05568, 2016.

[62] Edward Choi, Mohammad Taha Bahadori, Le Song, Walter F Stewart, and Jimeng Sun. Gram: graph-based attention model for healthcare representation learning. In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 787–795, 2017.

[63] Edward Choi, Mohammad Taha Bahadori, Jimeng Sun, Joshua Kulas, Andy Schuetz, and Walter Stewart. Retain: An interpretable predictive model for healthcare using reverse time attention mechanism. In Advances in Neural Information Processing Systems, pages 3504–3512, 2016.

[64] Edward Choi, Andy Schuetz, Walter F Stewart, and Jimeng Sun. Using recurrent neural network models for early detection of heart failure onset. Journal of the American Medical Informatics Association, 24(2):361–370, 08 2016.

[65] Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. Gated feedback recurrent neural networks. In International conference on machine learning, pages 2067–2075, 2015.

[66] Peter Clark and Tim Niblett. The cn2 induction algorithm. Machine learning, 3(4):261–283, 1989.

[67] Oana Cocarascu and Francesca Toni. Combining deep learning and argumentative reasoning for the analysis of social media textual content using small data sets. Computational Linguistics, 44(4):833–858, 2018.

[68] Corinna Cortes, Mehryar Mohri, and Afshin Rostamizadeh. Learning non-linear combinations of kernels. In Advances in neural information processing systems, pages 396–404, 2009.

[69] Sylvie Coste-Marquis, Caroline Devred, and Pierre Marquis. Prudent semantics for argumentation frameworks. In 17th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'05), pages 5–pp. IEEE, 2005.

[70] Sylvie Coste-Marquis, Caroline Devred, and Pierre Marquis. Symmetric argumentation frameworks. In European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty, pages 317–328. Springer, 2005.

[71] Sylvie Coste-Marquis, Caroline Devred, and Pierre Marquis. Constrained argumentation frameworks. KR, 6:112–122, 2006.

[72] Mark Craven and Jude W. Shavlik. Using sampling and queries to extract rules from trained neural networks. In ICML, 1994.

[73] Mark Craven and Jude W Shavlik. Extracting tree-structured representations of trained networks. In Advances in neural information processing systems, pages 24–30, 1996.

[74] Mark W. Craven and Jude W. Shavlik. Extracting tree-structured representations of trained networks. In Proceedings of the 8th International Conference on Neural Information Processing Systems, NIPS'95, pages 24–30, Cambridge, MA, USA, 1995. MIT Press.

[75] Ricardo Cruz-Correia, P Vieira-Marques, Pedro Costa, Ana Ferreira, E Oliveira-Palhares, F Araújo, and A Costa-Pereira. Integration of hospital data using agent technologies–a case study. Ai Communications, 18(3):191–200, 2005.

[76] Cesare Cuspidi, Federico Paoletti, Marijana Tadic, Carla Sala, Raffaella Dell'Oro, Guido Grassi, and Giuseppe Mancia. American versus european hypertension guidelines: the case of white coat hypertension. American Journal of Hypertension, 2020.

[77] Kristijonas Cyras, Brendan Delaney, Denys Prociuk, Francesca Toni, Martin Chapman, Jesús Dominguez, and Vasa Curcin. Argumentation for explainable reasoning with conflicting medical recommendations. 2018.

[78] Darren Dancey, Zuhair A Bandar, and David McLean. Rule extraction from neural networks for medical domains. In The 2010 International Joint Conference on Neural Networks (IJCNN), pages 1–8. IEEE, 2010.

[79] Howard B Demuth, Mark H Beale, Orlando De Jess, and Martin T Hagan. Neural network design. Martin Hagan, 2014.

[80] Peter Diehl and Matthew Cook. Unsupervised learning of digit recognition using spike-timing-dependent plasticity. Frontiers in Computational Neuroscience, 9:99, 2015.

[81] Thomas G Dietterich. Ensemble methods in machine learning. In International workshop on multiple classifier systems, pages 1–15. Springer, 2000.

[82] Frank Dignum and Mark Greaves. Issues in agent communication. Springer, 2006.

[83] Frank PM Dignum and Gerard AW Vreeswijk. Towards a testbed for multi-party dialogues. In Workshop on Agent Communication Languages, pages 212–230. Springer, 2003.

[84] Jeff Donahue, Lisa Anne Hendricks, Marcus Rohrbach, Subhashini Venugopalan, Sergio Guadarrama, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. IEEE Trans. Pattern Anal. Mach. Intell., 39(4):677–691, April 2017.

[85] Finale Doshi-Velez, Yaorong Ge, and Isaac Kohane. Comorbidity clusters in autism spectrum disorders: an electronic health record time-series analysis. Pediatrics, 133(1):e54–e63, 2014.

[86] Harris Drucker, Corinna Cortes, Lawrence D Jackel, Yann LeCun, and Vladimir Vapnik. Boosting and other ensemble methods. Neural Computation, 6(6):1289–1301, 1994.

[87] Giselle Dugelay, Joëlle Kivits, Louise Desse, and Jean-Marc Boivin. Implementation of home blood pressure monitoring among french gps: A long and winding road. PloS one, 14(9):e0220460, 2019.

[88] Phan Minh Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. Artificial Intelligence, 77(2):321 – 357, 1995.

[89] Phan Minh Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. Artif. Intell., 77(2):321–357, September 1995.

[90] Phan Minh Dung, Paolo Mancarella, and Francesca Toni. Computing ideal sceptical argumentation. Artificial Intelligence, 171(10-15):642–674, 2007.

[91] Edmund H Durfee and Thomas A Montgomery. A hierarchical protocol for coordinating mulitagent behaviors. In AAAI, pages 86–93, 1990.

[92] Shahram Ebadollahi, Jimeng Sun, David Gotz, Jianying Hu, Daby Sow, and Chalapathy Neti. Predicting patient's trajectory of physiological data using temporal trends in similar patients: a system for near-term prognostics. In AMIA annual symposium proceedings, volume 2010, page 192. American Medical Informatics Association, 2010.

[93] Brent M Egan, Jiexiang Li, Florence N Hutchison, and Keith C Ferdinand. Hypertension in the united states, 1999 to 2012: progress toward healthy people 2020 goals. Circulation, 130(19):1692–1699, 2014.

[94] Jeffrey L. Elman. Distributed representations, simple recurrent networks, and grammatical structure. Mach. Learn., 7(2–3):195–225, September 1991.

[95] Abdessamad Elrharras, S El Moukhlis, Rachid Saadane, Mohamed Wahbi, and A Hamdoun. Fpga-based fully parallel pca-ann for spectrum sensing. Computer and Information Science, 8(1):108, 2015.

[96] Morten Elvang-Gøransson, Paul J Krause, and John Fox. Acceptability of arguments as 'logical uncertainty'. In European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty, pages 85–90. Springer, 1993.

[97] Wael Farhan, Zhimu Wang, Yingxiang Huang, Shuang Wang, Fei Wang, and Xiaoqian Jiang. A predictive model for medical events based on contextual embedding of temporal sequences. JMIR Med Inform, 4(4):e39, Nov 2016.

[98] Wael Farhan, Zhimu Wang, Yingxiang Huang, Shuang Wang, Fei Wang, and Xiaoqian Jiang. A predictive model for medical events based on contextual embedding of temporal sequences. JMIR Medical Informatics, 4:e39, 11 2016.

[99] Hassan Ismail Fawaz, Germain Forestier, Jonathan Weber, Lhassane Idoumghar, and Pierre-Alain Muller. Deep learning for time series classification: a review. Data Mining and Knowledge Discovery, 33(4):917–963, 2019.

[100] Jacques Ferber. Multi-agent systems: an introduction to distributed artificial intelligence, volume 199. Addison-Wesley Reading, 1995.

[101] Jacques Ferber. Les systèmes multi-agents: un aperçu général. Techniques et sciences informatiques, 16(8), 1997.

[102] Sergio Pajares Ferrando and Eva Onaindia. Context-aware multi-agent planning in intelligent environments. Information Sciences, 227:22–42, 2013.

[103] Thomas Fischer and Christopher Krauss. Deep learning with long short-term memory networks for financial market predictions. FAU Discussion Papers in Economics 11/2017, Friedrich-Alexander University Erlangen-Nuremberg, Institute for Economics, 2017.

[104] A Fraisse and G Habib. Traitement de l'hypertension artérielle pulmonaire de l'enfant. Archives de pédiatrie, 11(8):945–950, 2004.

[105] Stan Franklin. Autonomous agents as embodied ai. Cybernetics & Systems, 28(6):499–520, 1997.

[106] Yoav Freund and Robert E. Schapire. Experiments with a new boosting algorithm, 1996.

[107] LiMin Fu. Rule generation from neural networks. 24:1114 – 1124, 09 1994.

[108] Sarah Alice Gaggl and Stefan Woltran. cf2 semantics revisited. In COMMA, pages 243–254, 2010.

[109] Alejandro J García and Guillermo R Simari. Defeasible logic programming: An argumentative approach. Theory and practice of logic programming, 4(1+ 2):95–138, 2004.

[110] GAËL. 3 algorithmes de deep learning expliqués en langage humain. 2018.

[111] JM Geleijnse, DE Grobbee, and FJ Kok. Impact of dietary and lifestyle factors on the prevalence of hypertension in western populations. Journal of human hypertension, 19(3):S1–S4, 2005.

[112] MR Genesereth and SP Ketchpel. Sp 1994. Software Agents.

[113] Kara Gilbert and Gordon Whyte. Argument and medicine: A model of reasoning for clinical practice. 2009.

[114] Earl F Glynn and Mark A Hoffman. Heterogeneity introduced by ehr system implementation in a de-identified data resource from 100 non-affiliated organizations. JAMIA open, 2(4):554–561, 2019.

[115] Benjamin A Goldstein, Ann Marie Navar, Michael J Pencina, and John P A Ioannidis. Opportunities and challenges in developing risk prediction models with electronic health records data: a systematic review. Journal of the American Medical Informatics Association, 24(1):198–208, 05 2016.

[116] Horacio González-Vélez, Mariola Mier, Margarida Julià-Sapé, Theodoros N Arvanitis, Juan M García-Gómez, Montserrat Robles, Paul H Lewis, Srinandan Dasmahapatra, David Dupplaw, Andrew Peet, et al. Healthagents: distributed multi-agent brain tumor diagnosis and prognosis. Applied intelligence, 30(3):191–202, 2009.

[117] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. Deep learning. MIT press, 2016.

[118] Ian J. Goodfellow, Yoshua Bengio, and Aaron Courville. Deep Learning. MIT Press, Cambridge, MA, USA, 2016.

[119] Henry Gouk, Eibe Frank, Bernhard Pfahringer, and Michael Cree. Regularisation of neural networks by enforcing lipschitz continuity. arXiv preprint arXiv:1804.04368, 2018.

[120] John W Graham. Missing data analysis: Making it work in the real world. Annual review of psychology, 60:549–576, 2009.

[121] Maria Adela Grando, David Glasspool, and Aziz Boxwala. Argumentation logic for the flexible enactment of goal-based medical guidelines. Journal of biomedical informatics, 45(5):938–949, 2012.

[122] Audrunas Gruslys, Remi Munos, Ivo Danihelka, Marc Lanctot, and Alex Graves. Memory-efficient backpropagation through time. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, Advances in Neural Information Processing Systems 29, pages 4125–4133. Curran Associates, Inc., 2016.

[123] Prakken H. Models of Persuasion Dialogue, pages 34–41. Simari G., Rahwan I. (eds) Argumentation in Artificial Intelligence. Springer, Boston, MA, 2009.

[124] Ivan Habernal and Iryna Gurevych. What makes a convincing argument? empirical analysis and detecting attributes of convincingness in web argumentation. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pages 1214–1223, 2016.

[125] Zhiyong Hao, Li Yao, Bin Liu, and Yanjuan Wang. Arguing Prism: An Argumentation Based Approach for Collaborative Classification in Distributed Environments, pages 34–41. Springer International Publishing, Cham, 2014.

[126] Peter Harrington. Machine Learning in Action. Manning Publications Co., USA, 2012.

[127] Salima Hassas. Systèmes complexes à base de multi-agents situés. University Claude Bernard Lyon, 2003.

[128] Yoichi Hayashi and Shota Fujisawa. Strategic approach for multiple-mlp ensemble re-rx algorithm. In International Joint Conference on Neural Networks (IJCNN'2015), pages 1–8, 07 2015.

[129] Barbara Hayes-Roth, Richard Washington, Rattikorn Hewett, Micheal Hewett, and Adam Seiver. Intelligent monitoring and control. In IJCAI, volume 89, pages 243–249. Citeseer, 1989.

[130] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. Neural Comput., 9(8):1735–1780, November 1997.

[131] Sepp Hochreiter, A. Steven Younger, and Peter R. Conwell. Learning to learn using gradient descent. In Proceedings of the International Conference on Artificial Neural Networks, ICANN '01, page 87–94, Berlin, Heidelberg, 2001. Springer-Verlag.

[132] Leroy Hood and Stephen H Friend. Predictive, personalized, preventive, participatory (p4) cancer medicine. Nature reviews Clinical oncology, 8(3):184–187, 2011.

[133] Eduardo R. Hruschka and Nelson F.F. Ebecken. Extracting rules from multilayer perceptrons in classification problems: A clustering-based approach. Neurocomputing, 70(1):384 – 397, 2006. Neural Networks.

[134] Anthony Hunter. A probabilistic approach to modelling uncertain logical arguments. International Journal of Approximate Reasoning, 54(1):47–81, 2013.

[135] Anthony Hunter and Matthew Williams. Aggregating evidence about the positive and negative effects of treatments. Artificial intelligence in medicine, 56(3):173–190, 2012.

[136] Costantino Iadecola, Kristine Yaffe, José Biller, Lisa C Bratzke, Frank M Faraci, Philip B Gorelick, Martha Gulati, Hooman Kamel, David S Knopman, Lenore J Launer, et al. Impact of hypertension on cognitive function: a scientific statement from the american heart association. Hypertension, 68(6):e67–e94, 2016.

[137] Sajid Iqbal, Wasif Altaf, Muhammad Aslam, Waqar Mahmood, and Muhammad Usman Ghani Khan. Application of intelligent agents in health-care. Artificial Intelligence Review, 46(1):83–112, 2016.

[138] David Isern and Antonio Moreno. A systematic literature review of agents applied in healthcare. Journal of medical systems, 40(2):43, 2016.

[139] L. C. Jain and L. R. Medsker. Recurrent Neural Networks: Design and Applications. CRC Press, Inc., Boca Raton, FL, USA, 1st edition, 1999.

[140] Hadassa Jakobovits and Dirk Vermeir. Robust semantics for argumentation frameworks. Journal of logic and computation, 9(2):215–261, 1999.

[141] Nick R Jennings. Commitments and conventions: The foundation of coordination in multi-agent systems. The knowledge engineering review, 8(3):223–250, 1993.

[142] Ulf Johansson, C Sonstrod, R Konig, and Lars Niklasson. Neural networks and rule extraction for prediction and explanation in the marketing domain. In Proceedings of the International Joint Conference on Neural Networks, 2003., volume 4, pages 2866–2871. IEEE, 2003.

[143] Michael I. Jordan. Supervised learning and systems with excess degrees of freedom. Technical report, USA, 1988.

[144] Ikram Chraibi Kaadoud, Nicolas P Rougier, and Frédéric Alexandre. Knowledge extraction from the learning of sequences in a long short term memory (lstm) architecture. arXiv preprint arXiv:1912.03126, 2019.

[145] Özgür Kafalı, Stefano Bromuri, Michal Sindlar, Tom van der Weide, Eduardo Aguilar Pelaez, Ulrich Schaechtle, Bruno Alves, Damien Zufferey, Esther Rodriguez-Villegas, Michael Ignaz Schumacher, et al. Commodity 12: A smart e-health environment for diabetes management. Journal of Ambient Intelligence and Smart Environments, 5(5):479–502, 2013.

[146] Deepak A Kaji, John R Zech, Jun S Kim, Samuel K Cho, Neha S Dangayach, Anthony B Costa, and Eric K Oermann. An attention based deep learning model of clinical events in the intensive care unit. PloS one, 14(2):e0211057, 2019.

[147] Antonis Kakas and Pavlos Moraitis. Argumentation based decision making for autonomous agents. In Proceedings of the second international joint conference on Autonomous agents and multiagent systems, pages 883–890, 2003.

[148] Antonis Kakas and Pavlos Moraitis. Adaptive agent negotiation via argumentation. In Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems, pages 384–391, 2006.

[149] Leila Kalankesh, James Weatherall, Thamer Ba-Dhfari, Iain E Buchan, and Andy Brass. Taming ehr data: using semantic similarity to reduce dimensionality. In MedInfo, pages 52–56, 2013.

[150] Prateek Karkare. Neural networks – an intuition. 2019.

[151] Faizal Khan and Omar Reyad. Application of intelligent multi agent based systems for e-healthcare security. arXiv preprint arXiv:2004.01256, 2020.

[152] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014.

[153] Arnfried U Klingbeil, Markus Schneider, Peter Martus, Franz H Messerli, and Roland E Schmieder. A meta-analysis of the effects of treatment on left ventricular mass in essential hypertension. The American journal of medicine, 115(1):41–46, 2003.

[154] Nadin Kokciyan, Isabel Sassoon, Anthony P Young, Martin Chapman, Talya Porat, C Ashworth, S Modgil, S Parsons, and E Sklar. Towards an argumentation system for supporting patients in self-managing their chronic conditions. AAAI, 2018.

[155] Gilad J Kuperman, Anne Bobb, Thomas H Payne, Anthony J Avery, Tejal K Gandhi, Gerard Burns, David C Classen, and David W Bates. Medication-related clinical decision support in computerized provider order entry systems: a review. Journal of the American Medical Informatics Association, 14(1):29–40, 2007.

[156] Imran Kurt, Mevlut Ture, and A Turhan Kurum. Comparing performances of logistic regression, classification and regression tree, and neural networks for predicting coronary artery disease. Expert systems with applications, 34(1):366–374, 2008.

[157] Bum Chul Kwon, Min-Je Choi, Joanne Taery Kim, Edward Choi, Young Bin Kim, Soonwook Kwon, Jimeng Sun, and Jaegul Choo. Retainvis: Visual analytics with interpretable and interactive recurrent neural networks on electronic medical records. IEEE transactions on visualization and computer graphics, 25(1):299–309, 2019.

[158] Isotta Landi, Benjamin S. Glicksberg, Hao-Chih Lee, Sarah Cherng, Giulia Landi, Matteo Danieletto, Joel T. Dudley, Cesare Furlanello, and Riccardo Miotto. Deep representation learning of electronic health records to unlock patient stratification at scale, 2020.

[159] Marek Laskowski, Bryan CP Demianyk, Julia Witt, Shamir N Mukhi, Marcia R Friesen, and Robert D McLeod. Agent-based modeling of the spread of influenza-like illness in an emergency department: a simulation study. IEEE Transactions on Information Technology in Biomedicine, 15(6):877–889, 2011.

[160] Simon Meyer Lauritsen, Mads Ellersgaard Kalør, Emil Lund Kongsgaard, Katrine Meyer Lauritsen, Marianne Johansson Jørgensen, Jeppe Lange, and Bo Thiesson. Early detection of sepsis utilizing deep learning on electronic health record event sequences, 2019.

[161] Steve Lawrence, C Lee Giles, Ah Chung Tsoi, and Andrew D Back. Face recognition: A convolutional neural-network approach. IEEE transactions on neural networks, 8(1):98–113, 1997.

[162] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. nature, 521(7553):436–444, 2015.

[163] Xiaozheng Li, Huazhen Wang, Huixin He, Jixiang Du, Jian Chen, and Jinzhun Wu. Intelligent diagnosis with chinese electronic medical records based on convolutional neural networks. BMC bioinformatics, 20(1):62, 2019.

[164] Fangzhen Lin and Yoav Shoham. Argument systems: A uniform basis for nonmonotonic reasoning. In KR, 1989.

[165] Zachary C. Lipton, David C. Kale, Charles Elkan, and Randall Wetzel. Learning to diagnose with lstm recurrent neural networks, 2015.

[166] Hongjun Lu, Rudy Setiono, and Huan Liu. Effective data mining using neural networks. IEEE Trans. on Knowl. and Data Eng., 8(6):957–961, December 1996.

[167] Scott Lundberg and Su-In Lee. An unexpected unity among methods for interpreting model predictions. arXiv preprint arXiv:1611.07478, 2016.

[168] Minh-Thang Luong, Hieu Pham, and Christopher D Manning. Effective approaches to attention-based neural machine translation. arXiv preprint arXiv:1508.04025, 2015.

[169] H. Tsukimoto M. Sato. Rule extraction from neural networks via decision tree induction. In International Joint Conference on Neural Networks (IJCNN '01), page 1870–1875, 2001.

[170] Fenglong Ma, Radha Chitta, Jing Zhou, Quanzeng You, Tong Sun, and Jing Gao. Dipole: Diagnosis prediction in healthcare via attention-based bidirectional recurrent neural networks. In Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining, pages 1903–1911, 2017.

[171] Xuezhe Ma and Eduard Hovy. End-to-end sequence labeling via bi-directional lstm-cnns-crf, 2016.

[172] Amina Maddouri, Nadia Abchiche-Mimouni, Samir Otmane, and Jalel Akaichi. Gaamaaa: Generating automatically an adaptive multiagent system for ambient assistive applications. In SoMeT, pages 39–54, 2019.

[173] Ivanny Marchant, El Manssouri Hanane, Behrouz Kassaï, Theodora Bejan-Angoulvant, Jacques Massol, Chrystelle Vidal, Emmanuel Amsallem, Florence Naudin, Pilar Galan, Sébastien Czernichow, Patrice Nony, and Francois Gueyffier. Score should be preferred to framingham to predict cardiovascular death in french population. 16:609–15, 10 2009.

[174] Nicolas Maudet, Simon Parsons, and Iyad Rahwan. Argumentation in multi-agent systems: Context and recent developments. In International Workshop on Argumentation in Multi-Agent Systems, pages 1–16. Springer, 2006.

[175] Tobias Mayer, Elena Cabrio, and Serena Villata. Acta: a tool for argumentative clinical trial analysis. 2019.

[176] Peter Mcburney and Simon Parsons. Dialogue games in multi-agent systems. Informal Logic, 22:2002, 2002.

[177] Manish Mehta, Rakesh Agrawal, and Jorma Rissanen. Sliq: A fast scalable classifier for data mining. In EDBT, 1996.

[178] Nir Menachemi and Taleah H Collum. Benefits and drawbacks of electronic health record systems. Risk management and healthcare policy, 4:47, 2011.

[179] Stéphane M Meystre, Guergana K Savova, Karin C Kipper-Schuler, and John F Hurdle. Extracting information from textual documents in the electronic health record: a review of recent research. Yearbook of medical informatics, 17(01):128–144, 2008.

[180] Tomas Mikolov, Martin Karafiát, Lukás Burget, Jan Cernocký, and Sanjeev Khudanpur. Recurrent neural network based language model. In Takao Kobayashi, Keikichi Hirose, and Satoshi Nakamura, editors, INTERSPEECH, pages 1045–1048. ISCA.

[181] Marvin Minsky and Doug Riecken. A conversation with marvin minsky about agents. Communications of the ACM, 37(7):22–29, 1994.

[182] Riccardo Miotto, Li Li, Brian A. Kidd, and Joel T Dudley. Deep patient: An unsupervised representation to predict the future of patients from the electronic health records. In Scientific reports, 2016.

[183] Riccardo Miotto and Chunhua Weng. Case-based reasoning using electronic health records efficiently identifies eligible patients for clinical trials. Journal of the American Medical Informatics Association, 22(e1):e141–e150, 2015.

[184] Thomas M. Mitchell. Machine Learning. McGraw-Hill, Inc., New York, NY, USA, 1 edition, 1997.

[185] Eiji Mizutani, Stuart E Dreyfus, and Kenichi Nishio. On derivation of mlp backpropagation from the kelley-bryson optimal-control gradient formula and its application. In Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium, volume 2, pages 167–172. IEEE, 2000.

[186] Sanjay Modgil. Hierarchical argumentation. In European Workshop on Logics in Artificial Intelligence, pages 319–332. Springer, 2006.

[187] Sanjay Modgil. Reasoning about preferences in argumentation frameworks. Artificial intelligence, 173(9-10):901–934, 2009.

[188] Sara Montagna, Stefano Mariani, Emiliano Gamberini, Alessandro Ricci, and Franco Zambonelli. Complementing agents with cognitive services: A case study in healthcare. Journal of Medical Systems, 44(10):1–10, 2020.

[189] Grégoire Montavon, Alexander Binder, Sebastian Lapuschkin, Wojciech Samek, and Klaus-Robert Müller. Layer-wise relevance propagation: an overview. In Explainable AI: interpreting, explaining and visualizing deep learning, pages 193–209. Springer, 2019.

[190] Milad Zafar Nezhad, Dongxiao Zhu, Xiangrui Li, Kai Yang, and Phillip Levy. Safs: A deep feature selection approach for precision medicine. 2016 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), Dec 2016.

[191] Andrew Ng et al. Sparse autoencoder. CS294A Lecture notes, 72(2011):1–19, 2011.

[192] Phuoc Nguyen, Truyen Tran, Nilmini Wickramasinghe, and Svetha Venkatesh. Deepr: A convolutional net for medical records. ArXiv, abs/1607.07519, 2016.

[193] Nils J Nilsson. Principles of artificial intelligence. Morgan Kaufmann, 2014.

[194] Paul D O'Brien and Richard C Nicol. Fipa—towards a standard for software agents. BT Technology Journal, 16(3):51–59, 1998.

[195] Ann Olincy, Josette G Harris, Lynn L Johnson, Vicki Pender, Susan Kongs, Diana Allensworth, Jamey Ellis, Gary O Zerbe, Sherry Leonard, Karen E Stevens, et al. Proof-of-concept trial of an $\alpha 7$ nicotinic agonist in schizophrenia. Archives of general psychiatry, 63(6):630–638, 2006.

[196] David Opitz and Richard Maclin. Popular ensemble methods: An empirical study. Journal of Artificial Intelligence Research, 11:169–198, 1999.

[197] Katie ATKINSON Trevor BENCH-CAPON Paul and E DUNNE. Uniform argumentation frameworks. Computational Models of Argument: Proceedings of COMMA 2012, 245:165, 2012.

[198] Gabriel Perlemuter and Léon Perlemuter. Guide pratique infirmier. Elsevier Masson, 2020.

[199] Trang Pham, Truyen Tran, Dinh Q. Phung, and Svetha Venkatesh. Deepcare: A deep dynamic memory model for predictive medicine. In PAKDD, 2016.

[200] Therese D Pigott. A review of methods for missing data. Educational research and evaluation, 7(4):353–383, 2001.

[201] Marty S Player and Lars E Peterson. Anxiety disorders, hypertension, and cardiovascular risk: a review. The International Journal of Psychiatry in Medicine, 41(4):365–377, 2011.

[202] John L. Pollock. Defeasible reasoning. Cognitive Science, 11(4):481 – 518, 1987.

[203] Jantima Polpinij, Natthakit Srikanjanapert, and Paphonput Sopon. Word2vec approach for sentiment classification relating to hotel reviews. In International Conference on Computing and Information Technology, pages 308–316. Springer, 2017.

[204] Henry Prakken. On dialogue systems with speech acts, arguments, and counterarguments. In European Workshop on Logics in Artificial Intelligence, pages 224–238. Springer, 2000.

[205] Henry Prakken. Coherence and flexibility in dialogue games for argumentation. Journal of logic and computation, 15(6):1009–1040, 2005.

[206] Henry Prakken. Formal systems for persuasion dialogue. Knowledge Engineering Review, 21(2):163, 2006.

[207] Henry Prakken. Models of persuasion dialogue. In Argumentation in artificial intelligence, pages 281–300. Springer, 2009.

[208] Henry Prakken. An abstract framework for argumentation with structured arguments. Argument and Computation, 1(2):93–124, 2010.

[209] Henry Prakken and Giovanni Sartor. A dialectical model of assessing conflicting arguments in legal reasoning. In Logical models of legal argumentation, pages 175–211. Springer, 1996.

[210] Henry Prakken and Giovanni Sartor. Argument-based extended logic programming with defeasible priorities. Journal of applied non-classical logics, 7(1-2):25–75, 1997.

[211] J. R. Quinlan. Induction of decision trees. Mach. Learn., 1(1):81–106, March 1986.

[212] J. Ross Quinlan. C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.

[213] Badran Raddaoui. Contributions aux approches logiques de l'argumentation en intelligence artificielle. 2013.

[214] Badran Raddaoui. Debate in a multi-agent system : multiparty argumentation protocols. 2016.

[215] Fatemeh Rahimian, Gholamreza Salimi-Khorshidi, Amir H Payberah, Jenny Tran, Roberto Ayala Solares, Francesca Raimondi, Milad Nazarzadeh, Dexter Canoy, and Kazem Rahimi. Predicting the risk of emergency admission with machine learning: Development and validation using linked electronic health records. PLoS medicine, 15(11):e1002695, 2018.

[216] Iyad Rahwan. Guest editorial: Argumentation in multi-agent systems. Autonomous Agents and Multi-Agent Systems, 11(2):115–125, 2005.

[217] Iyad Rahwan and Guillermo R Simari. Argumentation in artificial intelligence, volume 47. Springer, 2009.

[218] Laura Elena Raileanu and Kilian Stoffel. Theoretical comparison between the gini index and information gain criteria. Annals of Mathematics and Artificial Intelligence, 41(1):77–93, 2004.

[219] Alvin Rajkomar, Eyal Oren, Kai Chen, Andrew M Dai, Nissan Hajaj, Michaela Hardt, Peter J Liu, Xiaobing Liu, Jake Marcus, Mimi Sun, et al. Scalable and accurate deep learning with electronic health records. NPJ Digital Medicine, 1(1):18, 2018.

[220] H. Ramchoun, M. A. Janati Idrissi, Y. Ghanou, and M. Ettaouil. Multilayer perceptron: Architecture optimization and training with mixed activation functions. In Proceedings of the 2nd International Conference on Big Data, Cloud and Applications, BDCA'17, New York, NY, USA, 2017. Association for Computing Machinery.

[221] Rajesh Ranganath, Adler J. Perotte, Noémie Elhadad, and David M. Blei. The survival filter: Joint survival analysis with a latent time series. In UAI, 2015.

[222] Chris Reed. Representing dialogic argumentation. Knowledge-Based Systems, 19:22–31, 03 2006.

[223] Chris Reed and Douglas Walton. Argumentation schemes in argument-as-process and argument-as-product. 2003.

[224] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Introduction to local interpretable model-agnostic explanations (lime) a technique to explain the predictions of any machine learning classifier, 2016.

[225] Tjitze Rienstra, Alan Perotti, Serena Villata, Dov M Gabbay, and Leendert van der Torre. Multi-sorted argumentation. In International Workshop on Theorie and Applications of Formal Argumentation, pages 215–231. Springer, 2011.

[226] Marcela D Rodríguez, Jesus Favela, Alfredo Preciado, and Aurora Vizcaíno. Agent-based ambient intelligence for healthcare. Ai Communications, 18(3):201–216, 2005.

[227] S Russell and P Norvig. Artificial intelligence: a modern approach, 3rd edn london. UK: Pearson Education.[Google Scholar], 2014.

[228] David L Sackett. Evidence-based medicine. In Seminars in perinatology, volume 21, pages 3–5. Elsevier, 1997.

[229] Najibesadat Sadati, Milad Zafar Nezhad, Ratna Babu Chinnam, and Dongxiao Zhu. Representation learning with autoencoders for electronic health records: A comparative study, 2019.

[230] Omer Sagi and Lior Rokach. Ensemble learning: A survey. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 8(4):e1249, 2018.

[231] Sumit Saha. A comprehensive guide to convolutional neural networks — the eli5 way. 2018.

[232] Haşim Sak, Andrew Senior, and Françoise Beaufays. Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition, 2014.

[233] SAMUEL.LAURENCE.SMITH. Helping doctors validate decision trees. 2017.

[234] Emanuel Santos and João Pavão Martins. A default logic based framework for argumentation. In Proceedings of the 2008 conference on ECAI 2008: 18th European Conference on Artificial Intelligence, pages 859–860, 2008.

[235] João Santos, Joel JPC Rodrigues, Bruno MC Silva, João Casal, Kashif Saleem, and Victor Denisov. An iot-based mobile gateway for intelligent personal assistants on mobile health environments. Journal of Network and Computer Applications, 71:194–204, 2016.

[236] Makoto Sato and Hiroshi Tsukimoto. Rule extraction from neural networks via decision tree induction. In IJCNN'01, volume 3, pages 1870 – 1875 vol.3, 02 2001.

[237] Joseph L Schafer and John W Graham. Missing data: our view of the state of the art. Psychological methods, 7(2):147, 2002.

[238] Robert E Schapire. A brief introduction to boosting. In Ijcai, volume 99, pages 1401–1406, 1999.

[239] Bernhard Scholkopf, Kah-Kay Sung, Christopher JC Burges, Federico Girosi, Partha Niyogi, Tomaso Poggio, and Vladimir Vapnik. Comparing support vector machines with gaussian kernels to radial basis function classifiers. IEEE transactions on Signal Processing, 45(11):2758–2765, 1997.

[240] Nicholas J Schork. Personalized medicine: time for one-person trials. Nature, 520(7549):609–611, 2015.

[241] J. Searle. Speech acts. an essay in the philosophy of language. Cambridge University Press, 1969.

[242] John R Searle and John Rogers Searle. Speech acts: An essay in the philosophy of language, volume 626. Cambridge university press, 1969.

[243] Guillaume Vauvert-Amal El Fallah Seghrouchni. Formation de coalition pour agents rationnels.

[244] Rudy Setiono. Extracting rules from pruned neural networks for breast cancer diagnosis. Artificial intelligence in medicine, 8(1):37–51, 1996.

[245] Ying Sha and May D Wang. Interpretable predictions of clinical outcomes with an attention-based recurrent neural network. In Proceedings of the 8th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics, pages 233–240, 2017.

[246] John C. Shafer, Rakesh Agrawal, and Manish Mehta. Sprint: A scalable parallel classifier for data mining. In Proceedings of the 22th International Conference on Very Large Data Bases, VLDB '96, page 544–555, San Francisco, CA, USA, 1996. Morgan Kaufmann Publishers Inc.

[247] Lloyd S Shapley and Martin Shubik. A method for evaluating the distribution of power in a committee system. The American Political Science Review, 48(3):787–792, 1954.

[248] Onn M Shehory, Katia Sycara, and Somesh Jha. Multi-agent coordination through coalition formation. In International Workshop on Agent Theories, Architectures, and Languages, pages 143–154. Springer, 1997.

[249] Alex Sherstinsky. Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network. Physica D: Nonlinear Phenomena, 404:132306, Mar 2020.

[250] Benjamin Shickel, Tyler J Loftus, Lasith Adhikari, Tezcan Ozrazgat-Baslanti, Azra Bihorac, and Parisa Rashidi. Deepsofa: a continuous acuity score for critically ill patients using clinically interpretable deep learning. Scientific reports, 9(1):1–12, 2019.

[251] Yoav Shoham. Agent-oriented programming. Artificial intelligence, 60(1):51–92, 1993.

[252] Barry G Silverman, Nancy Hanrahan, Gnana Bharathy, Kim Gordon, and Dan Johnson. A systems approach to healthcare: agent-based modeling, community mental health, and population well-being. Artificial intelligence in medicine, 63(2):61–71, 2015.

[253] Guillermo R Simari and Ronald P Loui. A mathematical treatment of defeasible reasoning and its implementation. Artificial intelligence, 53(2-3):125–157, 1992.

[254] Shailendra Singh, Bukhary Ikhwan Ismail, Fazilah Haron, and Chan Huah Yong. Architecture of agent-based healthcare intelligent assistant on grid environment. In International Conference on Parallel and Distributed Computing: Applications and Technologies, pages 58–61. Springer, 2004.

[255] Stuart D Smith. Wind stress and heat flux over the ocean in gale force winds. Journal of Physical Oceanography, 10(5):709–726, 1980.

[256] Ingo Steinwart and Andreas Christmann. Support Vector Machines. Springer Publishing Company, Incorporated, 1st edition, 2008.

[257] Chuan-Jun Su and Ta-Wei Chu. A mobile multi-agent information system for ubiquitous fetal monitoring. International journal of environmental research and public health, 11(1):600–625, 2014.

[258] Q. Suo, F. Ma, Y. Yuan, M. Huai, W. Zhong, A. Zhang, and J. Gao. Personalized disease prediction using a cnn-based similarity learning method. In 2017 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), pages 811–816, 2017.

[259] Qiuling Suo, Fenglong Ma, Ye Yuan, Mengdi Huai, Weida Zhong, Aidong Zhang, and Jing Gao. Personalized disease prediction using a cnn-based similarity learning method. In 2017 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), pages 811–816. IEEE, 2017.

[260] Chenhao Tan, Vlad Niculae, Cristian Danescu-Niculescu-Mizil, and Lillian Lee. Winning arguments: Interaction dynamics and persuasion strategies in good-faith online discussions. In Proceedings of the 25th international conference on world wide web, pages 613–624, 2016.

[261] Matthias Thimm and Alejandro Javier García. Classification and strategical issues of argumentation games on structured argumentation frameworks. In AAMAS, pages 1247–1254, 2010.

[262] Robert J Tibshirani and Bradley Efron. An introduction to the bootstrap. Monographs on statistics and applied probability, 57:1–436, 1993.

[263] Pancho Tolchinsky, Ulises Cortes, Sanjay Modgil, Francisco Caballero, and Antonio Lopez-Navidad. Increasing human-organ transplant availability: Argumentation-based agent deliberation. IEEE Intelligent Systems, 21(6):30–37, 2006.

[264] Stephen E Toulmin. The uses of argument. Cambridge university press, 2003.

[265] Son N. Tran and Artur d'Avila Garcez. Knowledge extraction from deep belief networks for images. In IJCAI-2013 workshop on neural-symbolic learning and reasoning, 2013.

[266] M. Tuţescu. L'argumentation: introduction à l'étude du discours. Colecția Științe ale limbajului. Ed. Universităţii din Bucureşti, 1998.

[267] Félicien Vallet. Automatic structuring of tv talk show programs. 09 2011.

[268] Frans H. van Eemeren, Rob Grootendorst, A. Francisca Snoeck Henkemans, J. Anthony Blair, Ralph H. Johnson, Erik C. W. Krabbe, Christian Plantin, Douglas N. Walton, Charles A. Willard, John Woods, and David Zarefsky. Fundamentals of Argumentation Theory: A Handbook of Historical Backgrounds and Contemporary Developments. Lawrence Erlbaum Associates, 1996.

[269] William Van Melle. Mycin: a knowledge-based consultation program for infectious disease diagnosis. International journal of man-machine studies, 10(3):313–322, 1978.

[270] Paolo Verdecchia, Giuseppe Schillaci, Claudia Borgioni, Antonella Ciucci, Ivano Zampi, Roberto Gattobigio, Nicola Sacchi, and Carlo Porcellati. White coat hypertension and white coat effect similarities and differences. American journal of hypertension, 8(8):790–798, 1995.

[271] Gerard A.W. Vreeswijk. Abstract argumentation systems. Artificial Intelligence, 90(1):225 – 279, 1997.

[272] B Waeber and F Feihl. Arterial hypertension. factors favoring long-term compliance with therapy. Revue Medicale Suisse, 3(93):22–24, 2007.

[273] Douglas N Walton. Argumentation schemes for presumptive reasoning. Psychology Press, 1996.

[274] Haishuai Wang, Zhicheng Cui, Yixin Chen, Michael Avidan, Arbi Ben Abdallah, and Alexander Kronzer. Predicting hospital readmission via cost-sensitive deep learning. IEEE/ACM transactions on computational biology and bioinformatics, 15(6):1968–1978, 2018.

[275] Maya Wardeh, Frans Coenen, and Trevor Bench-Capon. Multi-agent based classification using argumentation from experience. Autonomous Agents and Multi-Agent Systems, 25(3):447–474, Nov 2012.

[276] Lawrence L Weed. Medical records that guide and teach (concluded). Yearbook of Medical Informatics, 212:1, 1968.

[277] Matt Williams and Jon Williamson. Combining argumentation and bayesian nets for breast cancer prognosis. Journal of Logic, Language and Information, 15(1-2):155–178, 2006.

[278] Michael Wooldridge. Intelligent agents: The key concepts. In Vladimír Mařík, Olga Štěpánková, Hana Krautwurmová, and Michael Luck, editors, Multi-Agent Systems and Applications II, pages 3–43, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg.

[279] Michael J Wooldridge and Nicholas R Jennings. Intelligent agents: Theory and practice. The knowledge engineering review, 10(2):115–152, 1995.

[280] Adam Z Wyner and Wim Peters. On rule extraction from regulations. In JURIX, volume 11, pages 113–122, 2011.

[281] Junyi Xu, Li Yao, and Le Li. Argumentation based joint learning: A novel ensemble learning approach. PLOS ONE, 10:e0127281, 05 2015.

[282] Shi Yan. Understanding lstm and its diagrams. 2016.

[283] Qiao Yang and John S Shieh. A multi-agent prototype system for medical diagnosis. In 2008 3rd International Conference on Intelligent System and Knowledge Engineering, volume 1, pages 1265–1270. IEEE, 2008.

[284] Safoora Yousefi, Fatemeh Amrollahi, Mohamed Amgad, Chengliang Dong, Joshua E Lewis, Congzheng Song, David A Gutman, Sameer H Halani, Jose Enrique Velazquez Vega, Daniel J Brat, et al. Predicting clinical outcomes from large scale cancer genomic profiles with deep survival models. Scientific reports, 7(1):1–11, 2017.

[285] Ting Yu, Tony Jan, Simeon Simoff, and John Debenham. Incorporating prior domain knowledge into inductive machine learning. Unpublished doctoral dissertation Computer Sciences, 2007.

[286] Jure Zabkar, Martin Mozina, Jerneja Vidednik, and Ivan Bratko. Argument based machine learning in. In Computational Models of Argument: Proceedings of COMMA 2006, volume 144, page 59. IOS Press, 2006.

[287] Jinghe Zhang, Kamran Kowsari, James H Harrison, Jennifer M Lobo, and Laura E Barnes. Patient2vec: A personalized interpretable deep representation of the longitudinal electronic health record. IEEE Access, 6:65333–65346, 2018.

[288] Tianran Zhang, Muhao Chen, and Alex AT Bui. Diagnostic prediction with sequence-of-sets representation learning for clinical events. In International Conference on Artificial Intelligence in Medicine, pages 348–358. Springer, 2020.

[289] Yingchun Zhang, Haoyi Zhou, Jianxin Li, Wanlu Sun, and Yahong Chen. A time-sensitive hybrid learning model for patient subgrouping. In 2018 International Joint Conference on Neural Networks (IJCNN), pages 1–8. IEEE, 2018.

[290] B. Zhao, H. Lu, S. Chen, J. Liu, and D. Wu. Convolutional neural networks for time series classification. Journal of Systems Engineering and Electronics, 28(1):162–169, 2017.

[291] Di Zhao and Chunhua Weng. Combining pubmed knowledge and ehr data to develop a weighted bayesian network for pancreatic cancer prediction. Journal of biomedical informatics, 44(5):859–868, 2011.

[292] Jing Zhao, Aron Henriksson, Maria Kvist, Lars Asker, and Henrik Boström. Handling temporality of clinical events for drug safety surveillance. In AMIA Annual Symposium Proceedings, volume 2015, page 1371. American Medical Informatics Association, 2015.

[293] Zhi-Hua Zhou, Yuan Jiang, and Shi-Fu Chen. Extracting symbolic rules from trained neural network ensembles. AI Commun., 16(1):3–15, January 2003.

[294] Zhi-Hua Zhou, Yuan Jiang, and Shi-Fu Chen. Extracting symbolic rules from trained neural network ensembles. AI Commun., 16(1):3–15, May 2003.

[295] Ming-jun ZHU, Zhen-tao WANG, Xiao-qing SHI, Li-hua HAN, and Hong-ling FAN. Proof-based medicine and its application in the study of chinese medicine [j]. Henan Traditional Chinese Medicine, 6, 2001.

[296] Z. Zhu, C. Yin, B. Qian, Y. Cheng, J. Wei, and F. Wang. Measuring patient similarities via a deep architecture with medical concept embedding. In 2016 IEEE 16th International Conference on Data Mining (ICDM), pages 749–758, 2016.

[297] Jan Ruben Zilke, Eneldo Loza Mencía, and Frederik Janssen. Deepred – rule extraction from deep neural networks. In Toon Calders, Michelangelo Ceci, and Donato Malerba, editors, Discovery Science, pages 457–473, Cham, 2016. Springer International Publishing.

université PARIS-SACLAY

ÉCOLE DOCTORALE
Physique et ingénierie : électrons, photons, sciences du vivant (EOBE)

Title: Approche transparente basée sur l'apprentissage profond et l'argumentation multi-agents pour la gestion de l'hypertension

Mots clés: Apprentissage profond, méthodes d'ensemble, systèmes multiagents, argumentation, dossiers électroniques de santé, interprétabilité.

Résumé: L'hypertension est connue pour être l'une des principales causes de maladies cardiaques et d'accidents vasculaires cérébraux, tuant environ 7,5 millions de personnes dans le monde chaque année, principalement en raison de son diagnostic tardif. Afin de confirmer le diagnostic de l'hypertension, il est nécessaire de collecter des mesures médicales répétées. Une solution consiste à exploiter ces mesures et à les intégrer dans les dossiers électroniques de santé par des algorithmes d'apprentissage automatique.

Dans ce travail, nous nous sommes focalisés sur les méthodes d'ensemble qui combinent plusieurs algorithmes d'apprentissage automatique pour la classification. Ces modèles ont été largement utilisés pour améliorer les performances de classification d'un seul classifieur. Pour cela, des méthodes telles que Bagging et Boosting sont utilisées. Ces méthodes utilisent principalement le vote majoritaire ou pondéré pour intégrer les résultats des classifieurs. Cependant, un inconvénient majeur de ces approches est leur opacité, car elles ne fournissent pas d'explication des résultats et ne permettent pas une intégration préalable des connaissances. Comme nous utilisons l'apprentissage automatique dans le domaine de la santé considéré critique, l'explication des résultats de classification et la possibilité d'introduire de connaissances à priori dans le modèle appris deviennent une nécessité.

Afin de pallier ces faiblesses, nous introduisons une nouvelle méthode d'ensemble basée sur l'argumentation multiagents.

L'intégration de l'argumentation et de l'apprentissage automatique s'est avérée fructueuse et l'utilisation de l'argumentation est un moyen pertinent de combiner les classifieurs. En effet, l'argumentation peut imiter le processus décisionnel humain pour réaliser la résolution des conflits.

Notre idée est d'extraire automatiquement les arguments des modèles d'apprentissage automatique et de les combiner à l'aide de l'argumentation. Cela permet d'exploiter les connaissances internes de chaque classifieur, de fournir une explication des décisions et de faciliter l'intégration des connaissances à priori.

Dans cette thèse, les objectifs étaient multiples. Du point de vue de l'application médicale, l'objectif était de prédire le traitement de l'hypertension artérielle et la date de la prochaine visite chez le médecin. D'un point de vue scientifique, l'objectif était d'ajouter de la transparence à la méthode d'ensemble et d'injecter des connaissances à priori dans le système. Les contributions de la thèse sont diverses:

- Explication des prédictions;
- Intégration des connaissances internes de classification;
- Injection des connaissances du domaine;
- Amélioration de la précision des prédictions;

Les résultats démontrent que notre approche fournit efficacement des explications et de la transparence aux prédictions des méthodes d'ensemble et qui est capable d'intégrer des connaissances cliniques et des connaissances du domaine dans le système. De plus, elle améliore les performances du deep learning.

Title: Transparent approach based on deep learning and multiagent argumentation for hypertension management.

Keywords: Deep learning, ensemble methods, argumentation, Multiagent system, Electronic health records, Interpretability.

Abstract: Hypertension is known to be one of the leading causes of heart disease and stroke, killing around 7.5 million people worldwide every year, mostly because of its late diagnosis. In order to confirm the diagnosis of Hypertension, it is necessary to collect repeated medical measurements. One solution is to exploit these measurements and integrate them into Electronic Health Records by Machine Learning algorithms.

In this work, we focused on ensemble learning methods that combine several machine learning algorithms for classification. These models have been widely used to improve classification performance of a single classifier. For that purpose, methods such as Bagging and Boosting are used. These methods mainly use majority or weighted voting to integrate the results of the classifiers. However, one major drawback of these approaches is their opacity, as they do not provide results explanation and they do not allow prior knowledge integration. As we use machine learning for healthcare, the explanation of classification results and the ability to introduce domain and clinical knowledge inside the learned model become a necessity.

In order to overcome theses weaknesses, we introduced a new ensemble method based on multiagent argumentation.

The integration of argumentation and machine learning has been proven to be fruitful and the use of argumentation is a relevant way for combining the classifiers. Indeed, argumentation can imitate human decision-making process to realize resolution of conflicts.

Our idea is to automatically extract the arguments from ML models and combine them using argumentation. This allows to exploit the internal knowledge of each classifier, to provide an explanation for the decisions and to facilitate the integration of domain and clinical knowledge.

In this thesis, objectives were multiple. From the medical application point of view, the goal was to predict the treatment of Hypertension and the date of the next doctor visit. From the scientific point of view, the objectives were to add transparency to ensemble method and to inject domain and clinical knowledge. The contributions of the thesis are various:

- Explaining predictions;
- Integrating internal classification knowledge;
- Injecting domain and clinical knowledge;
- Improving predictions accuracy.

The results demonstrate that our method effectively provides explanations and transparency of the ensemble methods predictions and it is able to integrate domain and clinical knowledge into the system. Moreover, it improves the performance of existing machine learning algorithms.

Université Paris-Saclay
Espace Technologique / Immeuble Discovery
Route de l'Orme aux Merisiers RD 128 / 91190 Saint-Aubin, France