Interpretable Narrative Explanation for ML Predictors with LP: A Case Study for XAI

Roberta Calegari Giovanni Ciatto Jason Dellaluce Andrea Omicini
Dipartimento di Informatica – Scienza e Ingegneria (DISI)
ALMA MATER STUDIORUM–Università di Bologna, Italy
Email: roberta.calegari@unibo.it, giovanni.ciatto@unibo.it, jason.dellaluce@studio.unibo.it, andrea.omicini@unibo.it

***Abstract*—In the era of digital revolution, individual lives are going to cross and interconnect ubiquitous online domains and offline reality based on smart technologies—discovering, storing, processing, learning, analysing, and predicting from huge amounts of environment-collected data. *Sub-symbolic* techniques, such as deep learning, play a key role there, yet they are often built as black boxes, which are not inspectable, interpretable, explainable. New research efforts towards *explainable artificial intelligence* (XAI) are trying to address those issues, with the final purpose of building *understandable*, *accountable*, and *trustable* AI systems—still, seemingly with a long way to go.**

Generally speaking, while we fully understand and appreciate the power of sub-symbolic approaches, we believe that *symbolic* approaches to machine intelligence, once properly combined with sub-symbolic ones, have a critical role to play in order to achieve key properties of XAI such as *observability*, *interpretability*, *explainability*, *accountability*, and *trustability*. In this paper we describe an example of integration of symbolic and sub-symbolic techniques. First, we sketch a general framework where symbolic and sub-symbolic approaches could fruitfully combine to produce intelligent behaviour in AI applications. Then, we focus in particular on the goal of building a *narrative explanation* for ML predictors: to this end, we exploit the logical knowledge obtained translating decision tree predictors into logical programs.

***Index Terms*—XAI, logic programming, machine learning, symbolic vs. sub-symbolic**

I. INTRODUCTION

Artificial intelligence (AI), *machine learning* (ML), and *deep learning* (DL) are nowadays intertwined with a growing number of aspects of people's every day life [1], [2]. In fact, more and more decisions are delegated by humans to software agents whose intelligent behaviour is not the result of some skilled developer endowing it with some clever code, but rather the consequence the agents' capability of learning, planning, or inferring what to do from data—or, roughly speaking, their *artificial intelligence*.

For instance, banks and insurance companies have adopted ML and statistical methods since decades, in order to decide whether or not to grant a loan to a given customer, or to estimate the most profitable insurance plan for her. Similarly, ML has been employed in order to help doctors with their diagnoses, provided that a set of symptoms has been properly identified for a given patient; whereas statistical and probabilistic inference have been employed to test drugs, in order to prove them effective or safe. Furthermore, virtually any person, as a consumer of services and goods, lets a number of ML-trained agents decide or suggest what to buy, like, or read—as any consumer is likely to be profiled by most of the companies and organisations he/she has interacted.

In spite of the large adoption, intelligent agents whose behaviour is the result of automatic synthesis / learning procedures are difficult to trust for most people—in particular when people are not expert in the fields of computer or data sciences, AI, statistics. This is especially true for agents leveraging on machine or deep learning based techniques, often producing models whose internal behaviour is opaque and hard to explain for their developers too.

There, agents often tend to accumulate their knowledge into *black-box* predictive models which are trained through ML or DL. Broadly speaking, the "black-box" expression is used to refer to models where knowledge is not explicitly represented – such as in neural networks, support vector machines, or Hidden Markov Chains –, and it is therefore difficult, for humans, to understand what a black-box actually knows, or what leads to a particular decision.

Such difficulty in understanding black-boxes content and functioning is what prevents people from fully trusting – and thus accepting – them. In several contexts, such as the medical or financial ones, it is not sufficient for intelligent agents to output bare decisions, since, for instance, ethical and legal issues may arise. An *explanation* for each decision is therefore often desirable, preferable, or even required. For instance, applications dealing with personal data need to face the challenges of achieving valid consent for data use and protecting confidentiality, and addressing threats to privacy, data protection, and copyright. Those issues are particularly challenging in critical application scenarios such as healthcare, often involving the use of image (i.e., identifiable) data from children. While issues of data ownership, data security, and data access are important, other ethical issues may arise: since the diagnostic accuracy and value of the result is determined by the amount and quality of data used in model training, the first potential concern is to avoid algorithmic bias, which may lead to social discrimination and result in inequitable access to healthcare, just related to the provenience of the collected data [1], [3].

Furthermore, it may happen that black-boxes *silently* learn something wrong (e.g., Google image recognition software that classified black people as gorillas [4], [5]), or something right, but in a biased way (like the "background bias" problem, causing for instance husky images to be recognised only

because of their snowy background [6]). In such situations, explanations are expected to provide useful insights for black-box developers.

To tackle such trust issues, the *eXplainable Artificial Intelligence* (XAI) research field has recently emerged, and a comprehensive research road map has been proposed by DARPA [7], targeting the themes of explainability and interpretability in AI – and in particular ML – as a challenge of paramount importance in a world where AI is becoming more and more pervasively adopted. There, DARPA reviews the main approaches to make AI either more interpretable or *a posteriori* explainable, it categorise the many currently available techniques aimed at building meaningful interpretations or explanations for black-box models, it summarises the open problems and challenges, and it provides a successful reference framework for the researchers interested in the field.

The main idea behind XAI is to employ *explanators* [8] to provide easy to understand insights for a given black-box and its particular decisions. An explanator is any procedure producing a meaningful explanation for some human observer, by leveraging on any combination of *(i)* the black-box, *(ii)* its input data, or *(iii)* its decisions or predictions. To this end, we believe that symbolic approaches to machine intelligence – properly integrated with sub-symbolic approaches – may have a role to play in order to achieve key properties such as *interpretability*, *observability*, *explainability*, *accountability*, and *trustability*.

In this paper we focus on the specific problem of building a *narrative explanation* of ML techniques—thus positioning our contribution into the specific *Narrative Generation* DARPA category [7]. In particular, we first show a general framework where symbolic and sub-symbolic techniques are fruitfully combined to produce intelligent behaviour in AI applications. Then, we focus on the translation of ML predictors into logical knowledge with the aim to *(i)* infer new knowledge, *(ii)* reason and act accordingly, and *(iii)* build the narrative explanation of a decision output (or prediction).

To this end, we propose an automatic procedure aimed at translating a ML predictor – here in particular we consider the case of *decision trees* (DT) – into logical knowledge. We argue that, when the source DT has been trained over a set of real data in order to produce a predictor, the corresponding logic program may be employed to produce a narrative explanation for any given prediction.

Despite being mostly focused on DT, our proposal represent a first step towards a more general approach. In fact, DT have been proposed as a general means for *explaining* the behaviour of virtually any black-box model [9], [10].

Accordingly, the reminder of this paper is organised as follows. Section II briefly recalls the ML concepts and terminology used in the paper as well as the main research efforts in the field. Then Section III introduces our vision of a framework for the integration of symbolic and sub-symbolic techniques. Finally, Section IV discusses early experiments alongside the prototype implementation.

II. Context

Machine learning often produces black-box predictors based on opaque models, thus hiding their internal logic to the user. This hinders explainability, and represents both a practical and an ethical issue for ML. As a result, many research approaches in the XAI field aim at overcoming that crucial weakness, sometimes at the cost of trading off accuracy against interpretability. So, we first (Subsection II-B) summarise the state of the art as well as the goal of XAI, then (Subsection II-A) introduce some background notions to define the terminology adopted.

A. Background

Since several practical AI problems – such as image recognition, financial and medical decision support systems – can be reduced to *supervised* ML – which can be further grouped in terms of either *classification* or *regression* problems [11], [12] –, in the reminder of this paper we focus on this set of ML problems.

In those cases, a *learning algorithm* is commonly exploited to estimate the specific nature and shape of an unknown *prediction* function (or *predictor*) $p^* : \mathcal{X} \to \mathcal{Y}$, mapping each input vector $\mathbf{x}$ from a given input space $\mathcal{X}$ into a prediction from a given output space $\mathcal{Y}$. To do so, the learning algorithm takes into account a number N of examples in the form $(\mathbf{x}_i, \mathbf{y}_i)$ such that $\mathbf{x}_i \in X \subset \mathcal{X}$, $\mathbf{y}_i \in Y \subset \mathcal{Y}$, and $|X| \equiv |Y| \equiv N$. There, each $\mathbf{x}_i$ represents an instance of the input data for which the expected output value $\mathbf{y}_i$ is known or has already been estimated. Such sorts of ML problems are said to be "supervised" because the expected targets Y are available, whereas they are said to be "regression" problems if Y consists of continuous or numerable values, or "classification" problems if Y consists of categorical values.

The learning algorithm usually assumes $p^* \in \mathcal{P}$, for a given *family* $\mathcal{P}$ of predictors—meaning that the unknown prediction function exists, and it is from $\mathcal{P}$. The algorithm then *trains* a predictor $\hat{p} \in \mathcal{P}$ such that the value of a given loss function $\lambda : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ – computing the discrepancy among predicted and expected outputs – is minimal or reasonably low—i.e.: $\hat{p} = \underset{p \in \mathcal{P}}{\operatorname{argmin}} \left\{ \sum_{i=1}^{N} \lambda(\mathbf{y}_i,\ p(\mathbf{x}_i)) \right\}$.

Depending on the predictor family $\mathcal{P}$ of choice, the nature of the learning algorithm and the admissible shapes of $\hat{p}$ may vary dramatically, as well as the their *interpretability*. Even if the interpretability of predictor families is not a well-defined feature, most authors agree on the fact that some predictor families are *more interpretable than others* [13]—in the sense that it is easier for humans to understand the functioning and the predictions of the former ones. For instance, it is widely acknowledged that *generalized linear models* (GLM) are more interpretable than neural networks (NN), whereas *decision trees* (DT) [14] are among the most interpretable families [8]. DT can be considered more interpretable due to their construction: that is, recursively partitioning the input space $\mathcal{X}$ through a number of splits or *decisions* based on the input data X, in such a way that the prediction in each partition

is constant, and the loss w.r.t. Y is low, while keeping the amount of partitions low as well. Without affecting generality, we focus on the case of mono-dimensional classification – thus we write y instead of $\mathbf{y}$ –, since other cases can be easily reduced to this one. We further assume the input space $\mathcal{X}$ is N-dimensional, and let n_j be the meta-variable representing the name of the j^{th} dimension of $\mathcal{X}$.

Under such hypotheses, a DT predictor $p_T \in \mathcal{P}_{dt}$ assumes a *binary* tree T exists such that each node is either

- a *leaf*, carrying and representing a prediction, i.e. and assignment for y,
- an *internal* node, carrying and representing a *decision*, i.e. a formula in the form $(n_j \leq c)$—where c is a constant *threshold* chosen by the learning algorithm.

Each node ν inherits a partition $X_\nu \subseteq X$ of the original input data, from its parent. Since the root node ν_0 has no parent, it is assigned to the whole set of input data—i.e. $X_{\nu_0} \equiv X$. The decision carried by each internal node splits its X_ν into two disjoint parts – X_ν^L and X_ν^R – along the j^{th} dimension of $\mathcal{X}$. In particular, X_ν^L contains all the residual $x_i \in X_\nu$ such that $(x_i^j \leq c_\nu)$ – which are inherited by ν left child –, whereas X_ν^R contains all the residual $x_i \in X_\nu$ such that $x_i^j > c_\nu$—which are inherited by by ν right child. A leaf node l is created whenever a sequence of splits (i.e., a path from the tree root to the leaf parent) leads to a partition X_l which is (almost) *pure*—roughly, meaning that X_l (mostly) contains input data x_i for which the expected output is the same y_l. In this case, we say that the prediction carried by l is y_l. Assuming such a tree T exists, in order to classify some input data $\mathbf{x} \in \mathcal{X}$, the predictor p_T simply navigates the path $P = (\nu_0, \nu_1, \nu_2, \ldots, l)$ of T such that all decisions ν_k are matched by $\mathbf{x}$, then it outputs y_l.

B. XAI: The need for explanation and interpretable models

Since the adoption of interpretable predictors usually comes at cost of a lower potential in terms of predictive performance, *explanations* are the newly preferred way for providing understandable predictions without necessarily sacrificing accuracy. The idea, and the main goal of XAI is to create intelligible and understandable explanations for uninterpretable predictors *without* replacing or modifying them. Thus explanations are built through a number of heterogeneous techniques, broadly referred to as *explanators* [8]—just to cite some, *decision rules* [15], *feature importance* [16], saliency masks [17], sensitivity analysis [18], etc.

The state of the art for explainability currently recognises two main sorts of explanators, namely, either local or global. While *local* explanators attempt to provide an explanation for each particular prediction of a given predictor p, the *global* ones attempt to provide an explanation for the predictor p as a whole. In other words, local explanators provide an answer to the question "why does p predict $\mathbf{y}$ for the input $\mathbf{x}$?" – such as the LIME technique presented in [6] –, whereas global explanators provide an answer to the question "how does p build its predictions?"—such as decision rules.

In spite of the many approaches proposed to explain black boxes, some important scientific questions still remain unanswered. One of the most important open problems is that, until now, there is no agreement on what an explanation is. Indeed, some approaches adopt as explanation a set of rules, others a decision tree, others rely on visualisation techniques [8]. Moreover, recent works highlight the importance for an explanation to guarantee some properties, e.g., soundness, completeness, and compactness [8].

This is why our proposal aims at integrating sub-symbolic approaches with symbolic ones. To this end, DT can be exploited as an effective bridge between the symbolic and sub-symbolic realms. In fact, DT can be easily *(i)* built from an existing sub-symbolic predictor, and *(ii)* translated into symbolic knowledge – as it is shown in the reminder of this paper – thanks to their rule-based nature.

Decision trees are an interpretable family of predictors that have been proposed as a *global* means for explaining other, less interpretable, sorts of black-box predictors [9], [10]—such as neural networks [19]. The main idea behind such an approach is to build a DT approximating the behaviour of a given predictor, possibly, by only considering its inputs and its outputs. Such approximation essentially trades off predictive performance with interpretability. In fact, the structure of such a DT would then be used to provide useful insights concerning the original predictor inner functioning.

Describing the particular means for extracting DT from black-boxes is outside the scope of this paper. Given the vast literature on the topic – e.g., consider reading [8], [20] for an overview or [19], [21], [22] for a practical examples – we simply assume an extracted DT is available and it has an high *fidelity*—meaning that the loss in terms of predictive performance is low, w.r.t. the original black-box. In fact, whereas there exist several works focussing on how to synthesise DT out of black-box predictors, no attention is paid to merging them with symbolic approaches, which can play a key role in enhancing the interpretability and explainability of the system. In this paper we focus on such a matter.

We believe that a logical representation of DT may be interesting and enabling for further research directions. For instance, as far as explainability is concerned, we show how logic-translated DT can be used to both navigate the knowledge stored within the corresponding predictors – thus acting as *global* explanators –, and produce *narrative* explanations for their predictions—thus acting as *local* explanators. Note that the restriction on the DT representation makes it easy to map DT onto logical clauses, since DT are finite and with a limited expressivity (if / else conditions).

III. Vision

Many approaches to ML nowadays are increasingly focussing on *sub-symbolic* approaches – such as deep learning with neural networks [23] – and on how to make them work on the large scale. As promising as this may look – with the premise of potentially minimizing the engineering efforts needed – it is increasingly acknowledged that those

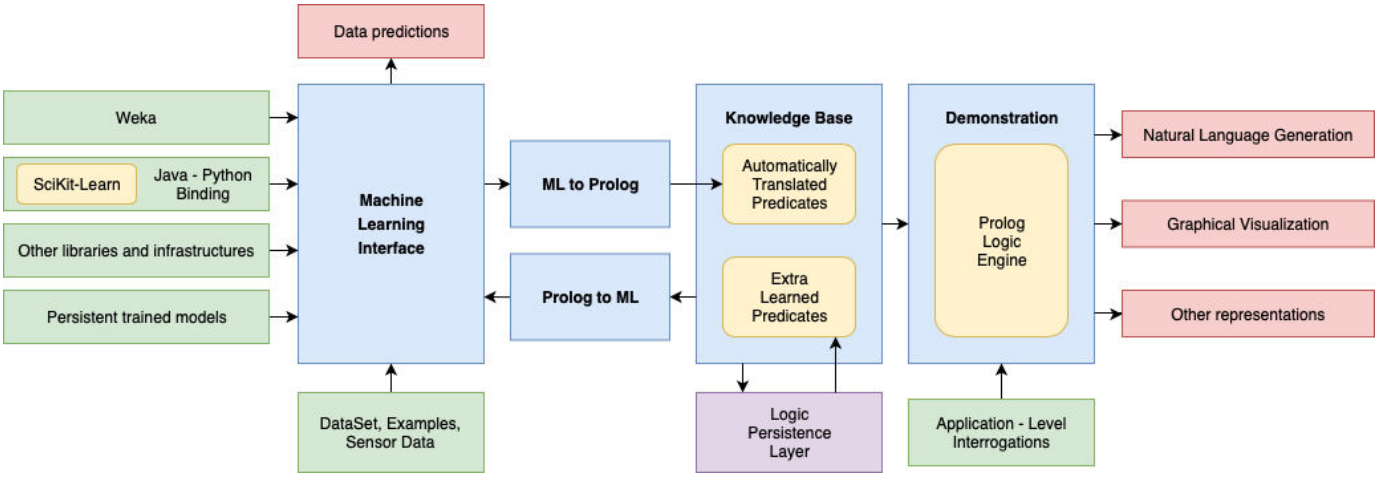

Fig. 1. ML to LP and back: framework architecture.

approaches do not cope well with the *socio-technical* nature of the systems they are exploited in, which often demand a degree of *interpretability*, *observability*, *explainability*, *accountability*, and *trustability* they just cannot deliver.

To this end, since logic-based approaches already have a well-understood role in building intelligent (multi-agent) systems [24], *declarative*, *logic-based* approaches have the potential to represent an alternative way of delivering symbolic intelligence, complementary to the one pursued by sub-symbolic approaches. In fact, declarative and logic-based technologies much better address the aforementioned socio-technical issues, in particular when exploiting their inferential capabilities—e.g., [25].

The potential of logic-based models and their extensions is first of all related to their declarativeness as well as to explicit knowledge representation, enabling knowledge sharing at the most adequate level of abstraction, while supporting modularity and separation of concerns [26]—which are especially valuable in open and dynamic distributed systems. As a further element, LP sound and complete semantics straightforwardly enables intelligent agents to reason and infer new information in a sound and complete way.

Another relevant point is that LP has been already proven to work well both as a knowledge representation language and as an inference platform for rational agents [27], [28]. The latter usually may interact with an external environment by means of a suitably defined observe–think–act cycle.

Accordingly to this vision, here we propose an integrated framework of hybrid reasoning – where symbolic and sub-symbolic techniques fruitfully combine to produce intelligent behaviour.

Indeed, looking in depth at pervasive socio-technical systems, it turns out that agents (either human or software) effortlessly undertake a complex decision making process in almost all situations, which seamlessly integrates perceptions (and actions) at two different scales—the *macro* and the *micro*:

- at the macro scale, by considering the knowledge of the *global* system, rules of general validity and concerning the most likely situation;
- at the micro scale, we modulate such decision by considering all the contingencies arising during the precise situation – such as, for instance, a last minute inconvenient, etc. As a consequence, we adapt the original plan to the *local* perceptions we gather while enacting it.

In order to better illustrate the above remarks, one may consider as a concrete example the case of a disease diagnosis in a hospital, where the notions of micro and macro scale w.r.t. to the nature of algorithms and techniques can be declined as follows:

- at the *macro level*, the main concerns regard a *mid/long term* horizon and focus the issue of analysis of high-dimensional and multimodal biomedical data train algorithms to recognize cancerous tissue at a level comparable to trained physicians—there including, for instance, representation and recognition of patterns and sequences in the input data. With such a sort of goals to pursue, it is not surprising that most IT tools supporting decision making are based on sub-symbolic approaches such as deep learning, Bayesian networks, machine vision, latent Dirichlet analysis, and in general any kind of statistical approach to ML [29], [30], [31]
- at the *micro level*, the main concerns regard instead the *short term* horizon, and mostly focus on the specific problem of the patient, there including a few highly-intertwined sub-problems—e.g. specific symptom or situation, ongoing epidemic in that hospital or place that carries the same symptoms. Although sub-symbolic approaches can still be used, *symbolic* ones such as fuzzy logic, specialized level (white box) learning instead of higher-level learning, symbolic time series are most common [29], [32], [33]

Generally speaking, we believe the computational intelligence accounts for this two kind of rules: *general rules* whose validity is essentially unconstrained (speed limits, right of way,

etc.) which represent the *commonsense knowledge* necessary to inhabit the environment and *specific rules*, with a validity bound in space and time (school hours and days, open-air market hours and days, unpredictable events such as incoming emergency vehicles the need to gather at an evacuation assembly point), which represent the *contextual* or expert knowledge necessary to deal with transient, unforeseen, and unpredictable situations.

That is why in the framework envisioned here we plan to combine sub-symbolic techniques with symbolic ones (LP in particular): sub-symbolic techniques are exploited for training the system and learn new rules (*commonsense knowledge*), rules are translated into logical knowledge (*contextual / expert knowledge*), and the two approaches interact and interleave to share knowledge and learn from each other in a coherent framework.

The framework architecture, depicted in Fig. 1, shows the embodiment of the vision discussed above: sensor data and dataset are translated into the logic knowledge base. In particular the *Machine Learning Interface* allows for the interaction of different kinds of ML algorithms with the framework: a standard interface is proposed in order to combine the specific features of each algorithm in a coherent manner. *ML to Prolog* is the core of the translation into logical knowledge, while the *Prolog to ML* returns insights of the logical KB to the ML predictor—for instance, new inferred rules, or rules learned by a specific situation. The blocks on the left (Knowledge Base, Demonstration) reflect the standard architecture of a Prolog engine. Overall, the framework looks general enough to account for the variety of ML techniques and algorithms, and also to ensure the consistency between symbolic and sub-symbolic approaches. Finally, the block *Prolog to ML* currently expresses our vision, and is obviously subject of future research.

IV. Early Experiments

The first prototype we design and implement enables the construction of a *narrative* explanation of the prediction generated exploiting the ML technique, thus achieving *interpretability* and making a step towards *explainability*.

With respect to Fig. 1, we experiment the predictor translation into logical rules, provided by the *ML to Prolog*. The experimental results refer to the case in which the predictor corresponds to a decision tree or to the corresponding crisp rules [34]. The conversion generates a Prolog predicate for each decision taken by the predictor: inside the predicate, a term for each input/output attribute is instantiated with the values of the leaf of the decision tree. A rule is generated for each leaf in the tree: between the other advantages, this allows for a very compact representation, easy to handle and interoperate with.

For a concrete example, let us consider the "Acute inflammations data set"[1] [35] supplying data to perform the presumptive diagnosis of two diseases of urinary system: the acute inflammations of urinary bladder and acute nephritises. Input parameters collect all the patient symptoms, each instance represents a potential patient. The data was created by a medical expert as a data set to test the expert system, which performs the presumptive diagnosis of two diseases of urinary system. The dataset considered is summarised in TABLE I and TABLE II.

TABLE I
Acute inflammations data set attributes

Attribute	Short name	Values
Temperature of patient	`temp`	$35°C \div 42°C$
Occurrence of nausea	`nausea`	{`yes`, `no`}
Lumbar pain	`lumbar`	{`yes`, `no`}
Urine pushing	`urine`	{`yes`, `no`}
Micturition pains	`micturition`	{`yes`, `no`}
Burning of urethra	`urethra`	{`yes`, `no`}
Output attributes		
Inflammation of urinary bladder	`inflammation`	{`yes`, `no`}
Nephritis of renal pelvis origin	`nephritis`	{`yes`, `no`}
Alternative output		
Diagnosis	`diagnosis`	{`healty`, `inflammation`, `nephritis`, `both`}

TABLE II
Acute inflammations data set description

Dataset size	120	
Num. of input attributes	6	
Num. of output attributes	2	
Num. of output classes	4	
Num. of healthy patients	30	(25%)
Num. of patients with inflammation of urinary bladder	59	(49.17%)
Num. of patients with nephritis of renal pelvis origin	50	(41.67%)
Num. of patients with both diseases	19	(15.83%)

Starting from the general form $Head \leftarrow Body$ for a logical clause, a predicate in the *Head* is generated for the decision of the predictor—in the example, the `diagnosis` predicate. Inside the predicate, a term for each input/output attribute is instantiated with the value of the decision tree (leaf).

In our example, the following predicate is generated:

```
diagnosis(temperatureOfPatient(T), occurrenceOfNausea(N),
     lumbarPain(L), urinePushing(U), micturitionPains(M),
     burningOfUrethra(BU), nephritisOfRenalPelvisOrigin(
     Decision), confidence(C)) :- Body.
```

where the *Body* body consists of check and computation on the variables of the *Head* terms. For instance, considering the above tree of Fig. 2, the first generated rule is

```
diagnosis(temperatureOfPatient(T), occurrenceOfNausea(N),
     lumbarPain(L), urinePushing(U), micturitionPains(M),
     burningOfUrethra(BU), nephritis(no), confidence(1.00))
     :- T =< 37.95.
```

[1] http://archive.ics.uci.edu/ml/datasets/acute+inflammations

representing the fact that if the temperature of patient is lesser or equal of 37.9, it is unlikely the patient presents nephritis of renal pelvis; the answer contains a degree of confidence based on the case of the dataset that confirm the rule—in the case 1.00 stands that all the patients in the dataset that have a temperature lower that 37.9 do not present the disease.

To improve readability, the rule above could be written as

```
diagnosis(temperatureOfPatient(T), _, _, _, _, _, nephritis
    (no), confidence(1.00)) :- T =< 37.95.
```

by omitting the undefined variables, i.e., highlighting the input attribute that are effectively to be considered as influencer.

Fig. 2 (left) depicts the whole picture: the decision trees generated as output of the example dataset when we run the basic classification tree algorithm[2] and the corresponding translation into LP rules. With respect to Fig. 1, the decision trees are the output of the *Machine Learning Interface* block and become the input for the *ML to Prolog* block.

Fig. 2 represents experiments of running the ML algorithm with no manipulation of the dataset: so, since the ML algorithm allows only one decision output to be considered for producing the corresponding decision tree, the information and the related knowledge is fragmented into two different trees – the first obtained running the algorithm with decision output *nephritis* and the second with decision output *inflammation of urinary bladder*. By running the *ML to Prolog* block of Fig. 1 we translate the two DT in LP rules as depicted in Fig. 2 (right).

a) Interpretabilty: The LP program provides an interpretable explanation of virtually any predictor. At a glance, the user can identify which attributes are meaningful and considered for response and which are not. In case of nephritis, the only significant input attributes are the temperature of patient and the presence or absence of lumbar pain. The same is for inflammation of urinary bladder, where the only discriminative attributes are presence of urine pushing, micturition pains and lumbar pain.

b) Interoperability: The adoption of a standard AI language (LP), in spite of the plethora of different specific ML toolkits, paves the way towards an interoperable explanation where LP is exploited as sort of *lingua franca* that goes beyond the technical implementation of each ML framework.

c) Relations between outputs: As emphasised by Fig. 2, relations between outputs are lost, and possible links between the diseases are not clearly highlighted having two different decision trees. Instead, once obtained a LP representation, it is easy to run simple queries on it in order to get much more information with respect to the two different decision tree. For instance, we can learn that in case of fever (temperature of patient $>$ 37.95) not presenting nephritis (i.e. no lumbar pain detected), the only case in which inflammation of urinary bladder is present is when urine pushing is detected in absence of symptoms of micturition pains. With the logical representation, relations between output can be recovered by inferring hidden knowledge in the rules. It is worth noticing that similar results (emphasising the relations between decision output) can be obtained manipulating the dataset *a priori*—i.e. before the ML algorithm training (a common operation but not always applicable). The manipulation of the above dataset, for instance, can build a unique decision output *Result* that combines the two different diseases and their symptoms. In such a case the dataset is enriched with the *Result* attribute containing the complete diagnosis, i.e., it can assume the values *Healthy, Inflammation, Nephritis, Both*. The corresponding decision tree and LP knowledge is depicted in Fig. 3.

d) Interpretable narrative explanation: LP makes it possible to generate a narration for each answer of the predictor. The inference Prolog tree becomes *inspectable*, tracking the path for obtaining the answer. For instance, w.r.t. the KB of Fig. 3 – including all diseases –, the diagnosis in the case of the following symptoms:

```
diagnosis(
      temperatureOfPatient(36.5), occurrenceOfNausea(yes),
      lumbarPain(yes), urinePushing(no),
      micturitionPains(yes), burningOfUrethra(yes), _, _).
```

would produce the corresponding narration:

```
The diagnosis is healthy, with a full confidence because
the patient has no fever.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
In particular the solution has been built across the
following path:
Solution: result(healthy) with confidence(1.00).

For the proof, the following clauses are considered:
[1] diagnosis(temperatureOfPatient(T), _, _, urinePushing(
     no), _, _, result(healthy), confidence(1.00)) :- T =< 3
     7.95.
[2] X =< Y that is verified if '
     expression_less_or_equal_than'(X, Y)

In the query the temperature T is of 36.5.
  because of rule [1]  36.5 =< 36.9 has to be verified
  and because of [2] 'expression_less_or_equal_than'(36.5,
     36.9) has to be verified
so rules [1]  and [2] are verified.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Despite its simplicity, the narration allows for a reconstruction of the decision track, showing the path to the decision. With a large amount of nested rules this could result very effective.

e) Exploitation of LP extension / abduction on the KB: Moreover, we believe that exploiting abduction techniques we could pave the way to hypothetical reasoning with incomplete knowledge, i.e., learning new possible hypotheses that can be assumed to hold, provided that they are consistent with the given knowledge base. The idea, to be explored in future research, is to provide the most likely solution given a set of evidence. The conclusion would leave a degree of uncertainty while highlighting a plausible answer based on the collected information. In the healthcare field, for instance, it could be represented by having the collection of symptoms (although incomplete) and finding the most likely disease for them.

[2]We exploit two different implementations: C45 [36] weka J48 for the Java translator and SciKit-Learn CART [14] for the Phyton one

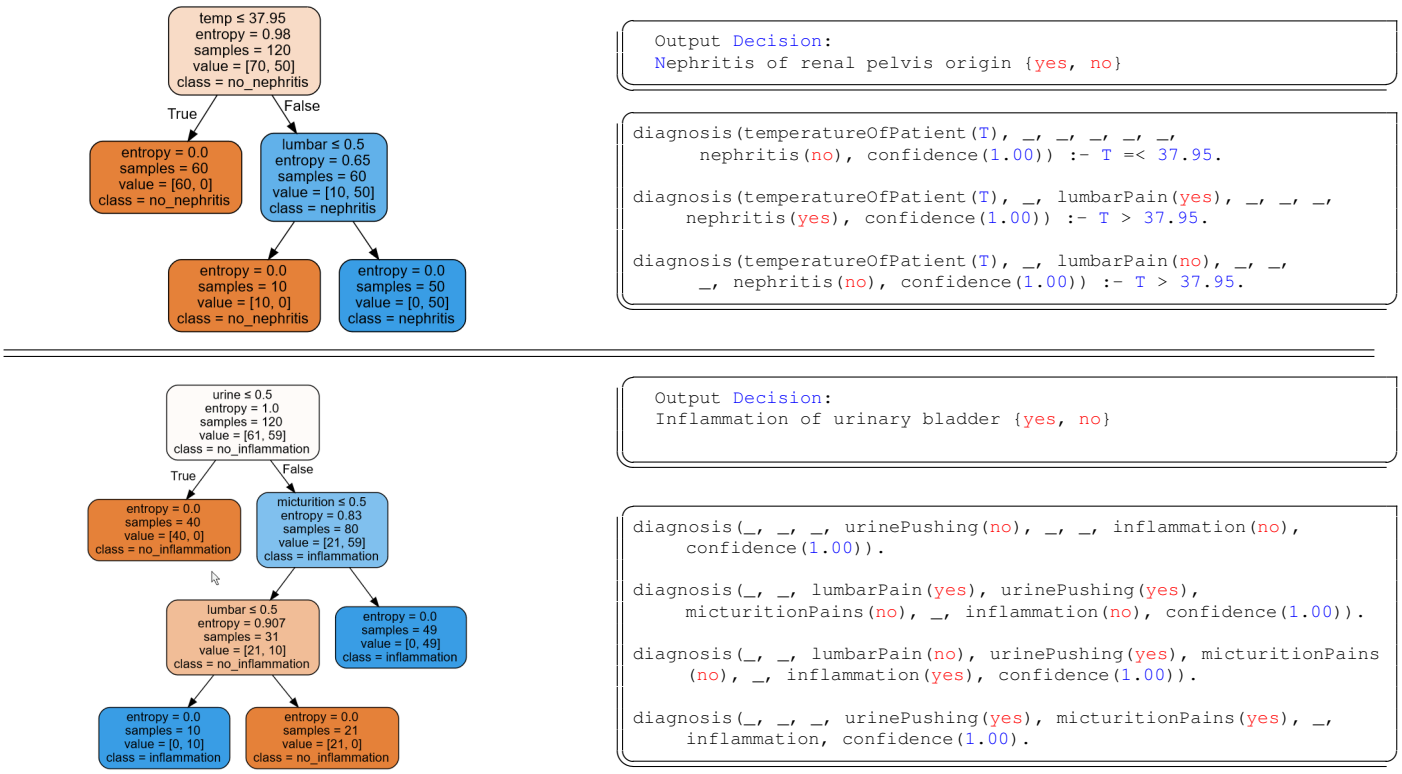

Fig. 2. Experimental results obtained running the framework on the Acute Inflammations dataset [35]: on the $left$ side are represented the decision trees generated by the supervised ML algorithm (Weka J48 – SciKit-Learn CART), while on the $right$ the corresponding LP rules output of the *ML to Prolog* block. In order to deal with two different overlapped outputs, two DT are generated: information are not connected as the knowledge.

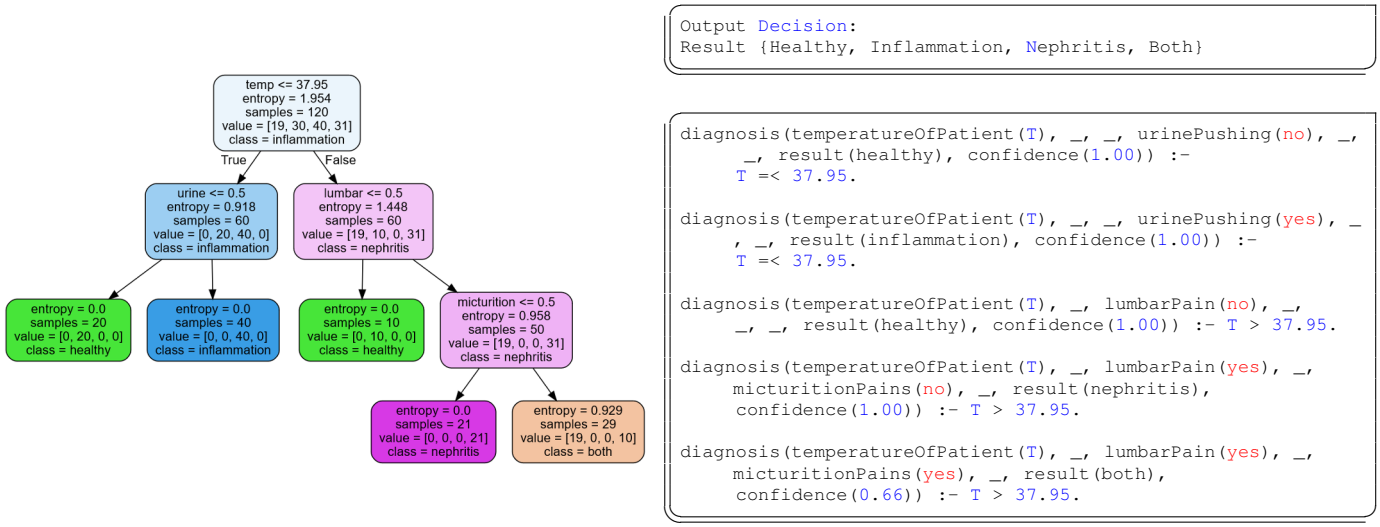

Fig. 3. Decision Tree (left) and corresponding *"ML to Prolog core"* output (right) after the previous manipulation of the dataset. In particular the two different output decisions (nephritis and inflammation of urinary bladder) have been combined in order to generate a comprehensive output decision: the new diagnosis consider that case of a healthy patient (none of the previous diseases), the case in which only one of the two diseases is present (inflammation or nephritis), and finally the case in which are both present.

V. Conclusion

AI systems nowadays synthesise large amounts of data, learning from experience and making predictions with the goal of taking autonomous decisions—applications range from clinical decision support to autonomous driving and predictive policing. Nevertheless, concerns about the intentional and unintentional negative consequences of AI systems are legitimate, as well as ethical and legal concerns, mostly related to darkness and opaqueness of AI decision algorithm. For that reason, recent work on interpretability in machine learning and

AI has focused on simplified models that approximate the true criteria used to make decisions.

In this paper we focus on building a narrative explanation of the machine learning techniques: we first translate a ML predictor into logical knowledge, then inspect the proof tree leading to a solution. The narration is built tracking the path (i.e., the rules) that leads from the query to the answer.

Along this line, we foresee a broader vision that involves the design of a consistent framework where symbolic and sub-symbolic techniques are fruitfully combined to produce intelligent behaviour in AI applications while exploiting the benefits of each approach—like, in the case of symbolic ones, interpretability, observability, explainability, and accountability.

The results presented here represent just a preliminary exploration of the potential benefits of merging symbolic and sub-symbolic approaches—where, of course, many critical issues are still unexplored and will be subject of future work. However, despite its simplicity, the case study already allows us to point out the feasibility and the potential benefits of the exploitation of symbolic techniques towards XAI.

REFERENCES

[1] D. Helbing, "Societal, economic, ethical and legal challenges of the digital revolution: From big data to deep learning, artificial intelligence, and manipulative technologies," in *Towards Digital Enlightenment*. Springer, 2019, pp. 47–72.

[2] A. Elliott, *The Culture of AI: Everyday Life and the Digital Revolution*. Routledge, 2019.

[3] S. Bird, K. Kenthapadi, E. Kiciman, and M. Mitchell, "Fairness-aware machine learning: Practical challenges and lessons learned," in *12th ACM International Conference on Web Search and Data Mining (WSDM'19)*. ACM, 2019, pp. 834–835.

[4] M. Fourcade and K. Healy, "Categories all the way down," *Historical Social Research/Historische Sozialforschung*, pp. 286–296, 2017.

[5] K. Crawford, "Artificial intelligence's white guy problem," *The New York Times*, vol. 25, 2016.

[6] M. T. Ribeiro, S. Singh, and C. Guestrin, "Why should I trust you? Explaining the predictions of any classifier," *CoRR*, vol. abs/1602.04938, 2016.

[7] D. Gunning, "Explainable artificial intelligence (XAI)," DARPA, Funding Program DARPA-BAA-16-53, 2016. [Online]. Available: http://www.darpa.mil/program/explainable-artificial-intelligence

[8] R. Guidotti, A. Monreale, F. Turini, D. Pedreschi, and F. Giannotti, "A survey of methods for explaining black box models," *CoRR*, vol. abs/1802.01933, 2018.

[9] F. Di Castro and E. Bertini, "Surrogate decision tree visualization," in *Joint Proceedings of the ACM IUI 2019 Workshops (ACMIUI-WS 2019)*, ser. CEUR Workshop Proceedings, vol. 2327, Mar. 2019.

[10] O. Bastani, C. Kim, and H. Bastani, "Interpreting blackbox models via model extraction," *CoRR*, vol. abs/1705.08504, 2017.

[11] B. Twala, "Multiple classifier application to credit risk assessment," *Expert Systems with Applications*, vol. 37, no. 4, pp. 3326–3336, 2010.

[12] S. Kotsiantis, "Supervised machine learning: A review of classification techniques," in *Emerging Artificial Intelligence Applications in Computer Engineering*, ser. Frontiers in Artificial Intelligence and Applications. IOS Press, Oct. 2007, vol. 160, pp. 3–24.

[13] Z. C. Lipton, "The mythos of model interpretability," *CoRR*, vol. abs/1606.03490, 2016.

[14] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees*. Chapman & Hall/CRC, 1984.

[16] G. Tolomei, F. Silvestri, A. Haines, and M. Lalmas, "Interpretable predictions of tree-based ensembles via actionable feature tweaking," in *23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2017, pp. 465–474. [Online]. Available: http://dl.acm.org/citation.cfm?id=3098039

[15] M. G. Augasta and T. Kathirvalavakumar, "Reverse engineering the neural networks for rule extraction in classification problems," *Neural Processing Letters*, vol. 35, no. 2, pp. 131–150, Apr. 2012.

[17] R. Fong and A. Vedaldi, "Interpretable explanations of black boxes by meaningful perturbation," *CoRR*, vol. abs/1704.03296, 2017.

[18] M. Sundararajan, A. Taly, and Q. Yan, "Axiomatic attribution for deep networks," *CoRR*, vol. abs/1703.01365, 2017.

[19] M. W. Craven and J. W. Shavlik, "Extracting tree-structured representations of trained networks," in *8th International Conference on Neural Information Processing Systems (NIPS'95)*. MIT Press, 1995, pp. 24–30.

[20] R. Andrews, J. Diederich, and A. B. Tickle, "Survey and critique of techniques for extracting rules from trained artificial neural networks," *Knowledge-Based Systems*, vol. 8, no. 6, pp. 373–389, Dec. 1995.

[21] U. Johansson and l. Niklasson, "Evolving decision trees using oracle guides," in *2009 IEEE Symposium on Computational Intelligence and Data Mining*, Mar. 2009, pp. 238–244.

[22] N. Frosst and G. E. Hinton, "Distilling a neural network into a soft decision tree," in *CEX 2017 Comprehensibility and Explanation in AI and ML 2017 (CEX 2017)*, ser. CEUR Workshop Proceedings, vol. 2071, Nov. 2017.

[23] D. Silver *et al.*, "Mastering the game of Go with deep neural networks and tree search," *Nature*, vol. 529, pp. 484–489, Jan. 2016.

[24] A. Omicini and F. Zambonelli, "MAS as complex systems: A view on the role of declarative approaches," in *Declarative Agent Languages and Technologies*, ser. Lecture Notes in Computer Science. Springer, May 2004, vol. 2990, pp. 1–17.

[25] F. Idelberger, G. Governatori, R. Riveret, and G. Sartor, "Evaluation of logic-based smart contracts for blockchain systems," in *Rule Technologies. Research, Tools, and Applications*, ser. Lecture Notes in Computer Science, vol. 9718. Springer, 2016, pp. 167–183.

[26] M. Oliya and H. K. Pung, "Towards incremental reasoning for context aware systems," in *Advances in Computing and Communications*, ser. Communications in Computer and Information Science. Springer, 2011, vol. 190, pp. 232–241.

[27] G. Sotnik, "The SOSIEL platform: Knowledge-based, cognitive, and multi-agent," *Biologically Inspired Cognitive Architectures*, vol. 26, pp. 103–117, Oct. 2018.

[28] R. Kowalski and F. Sadri, "From logic programming towards multi-agent systems," *Annals of Mathematics and Artificial Intelligence*, vol. 25, no. 3, pp. 391–419, Nov. 1999.

[29] M. D. Pandya, P. D. Shah, and S. Jardosh, "Medical image diagnosis for disease detection: A deep learning approach," in *U-Healthcare Monitoring Systems*, ser. Advances in Ubiquitous Sensing Applications for Healthcare. Academic Press, 2019, vol. 1: Design and Applications, ch. 3, pp. 37–60.

[30] S. Kuwayama, Y. Ayatsuka, D. Yanagisono, T. Uta, H. Usui, A. Kato, N. Takase, Y. Ogura, and T. Yasukawa, "Automated detection of macular diseases by optical coherence tomography and artificial intelligence machine learning of optical coherence tomography images," *Journal of Ophthalmology*, vol. 2019, p. 7, 2019.

[31] P. Sajda, "Machine learning for detection and diagnosis of disease," *Annual Review of Biomedical Engineering*, vol. 8, pp. 537–565, Aug. 2006.

[32] C. Zhang, Y. Chen, A. Yin, and X. Wang, "Anomaly detection in ECG based on trend symbolic aggregate approximation," *Mathematical Biosciences and Engineering*, vol. 16, no. 4, pp. 2154–2167, 2019.

[33] A. Rastogi, R. Arora, and S. Sharma, "Leaf disease detection and grading using computer vision technology & fuzzy logic," in *2nd International Conference on Signal Processing and Integrated Networks (SPIN 2015)*. IEEE, 2015, pp. 500–505.

[34] A. Lozowski, T. J. Cholewo, and J. M. Zurada, "Crisp rule extraction from perceptron network classifiers," in *IEEE International Conference on Neural Networks (ICNN 1996)*, vol. Plenary, Panel and Special Sessions, Jun. 1996, pp. 94–99.

[35] J. Czerniak and H. Zarzycki, "Application of rough sets in the presumptive diagnosis of urinary system diseases," in *Artificial Intelligence and Security in Computing Systems*, ser. The Springer International Series in Engineering and Computer Science. Springer, 2003, vol. 752, pp. 41–51.

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