ARCHIVIO ISTITUZIONALE
DELLA RICERCA

Alma Mater Studiorum Università di Bologna
Archivio istituzionale della ricerca

On the integration of symbolic and sub-symbolic techniques for XAI: A survey

This is the final peer-reviewed author's accepted manuscript (postprint) of the following publication:

Published Version:
On the integration of symbolic and sub-symbolic techniques for XAI: A survey / Roberta Calegari, Giovanni Ciatto, Andrea Omicini. - In: INTELLIGENZA ARTIFICIALE. - ISSN 1724-8035. - STAMPA. - 14:1(2020), pp. 7-32. [10.3233/IA-190036]

Availability:
This version is available at: https://hdl.handle.net/11585/772707 since: 2020-10-21

Published:

DOI: http://doi.org/10.3233/IA-190036

Terms of use:

Some rights reserved. The terms and conditions for the reuse of this version of the manuscript are specified in the publishing policy. For all terms of use and more information see the publisher's website.

This item was downloaded from IRIS Università di Bologna (https://cris.unibo.it/).
When citing, please refer to the published version.

(Article begins on next page)

20 April 2024

This is the final peer-reviewed accepted manuscript of:

Calegari, Roberta, Ciatto, Giovanni, and Omicini, Andrea. 'On the Integration of Symbolic and Sub-symbolic Techniques for XAI: A Survey'. 1 Jan. 2020 : 7 – 32.

Journal: Intelligenza Artificiale
Published: 17 September 2020

The final published version is available online at:

http://dx.doi.org/10.3233/IA-190036

Rights / License:

The terms and conditions for the reuse of this version of the manuscript are specified in the publishing policy. For all terms of use and more information see the publisher's website.

https://www.iospress.nl/service/authors/compliance-with-major-funding-agency-policies/

This item was downloaded from IRIS Università di Bologna (https://cris.unibo.it/)

When citing, please refer to the published version.

Intelligenza Artificiale 14 (2020) 7–32
DOI: 10.3233/IA-190036
IOS Press

On the integration of symbolic and sub-symbolic techniques for XAI: A survey

Roberta Calegari, Giovanni Ciatto and Andrea Omicini
Dipartimento di Informatica – Scienza e Ingegneria (DISI), ALMA MATER STUDIORUM–*Università di Bologna, Italy, E-mail: {roberta.calegari,giovanni.ciatto,andrea.omicini}@unibo.it*

Abstract.
The more intelligent systems based on sub-symbolic techniques pervade our everyday lives, the less human can understand them. This is why symbolic approaches are getting more and more attention in the general effort to make AI interpretable, explainable, and trustable. Understanding the current state of the art of AI techniques integrating symbolic and sub-symbolic approaches is then of paramount importance, nowadays—in particular in the XAI perspective. This is why this paper provides an overview of the main symbolic/sub-symbolic integration techniques, focussing in particular on those targeting explainable AI systems.

Keywords: XAI, symbolic and sub-symbolic AI, explainability, interpretability, trustable system

1. Introduction

In the era of the fourth industrial revolution, we are witnessing a fast and widespread adoption of AI in disparate domains. Nowadays, new AI applications often leverage on sub-symbolic approaches – e.g., deep learning [1,2,3] – to provide sophisticated features that would be hard for human developers to implement otherwise.

However, despite the unprecedented performance arising from the exploitation of sub-symbolic approaches, experts and practitioners nowadays widely acknowledge that state-of-the-art sub-symbolic models lack key features such as inspectability, interpretability, or explainability, given that they are inherently designed, trained, and used as black boxes [4]. In other words, sub-symbolic techniques perform very well in learning complex relationships and tasks from data, but they struggle in easing humans' understanding of how a particular relationship or task could or should be performed. This is potentially troublesome since, especially in critical domains – such as the medical, financial, or legal ones, where sub-symbolic intelligent systems are exploited to support or automate decision making –, ethical and legal issues are likely to arise. In fact, in all the cited contexts, it is not enough for intelligent systems to support human decisions: instead, they are also expected to provide motivations and explanations giving insights about *how* and *why* decisions are attained.

Broadly speaking, the "black box" expression is used to refer to models where knowledge is not explicitly represented, but rather it is *distributed* among tensors of real numbers—whose complexity seldom fits our cognitive capabilities as humans. This is why human beings can rarely understand what a black box actually knows, or how it achieved a particular decision. In turn, troubles in understanding black-box behaviour is what prevents people from fully trusting – and thus accepting – them. In other words, the "black box" expression stresses the intrinsically *opaque* nature of most sub-symbolic techniques—which is one of the main limitations of the current state of the art.

New research efforts towards *eXplainable artificial intelligence* (XAI) [5] are aimed at mitigating the opacity issue, and pursue the ultimate goal of building *understandable*, *accountable*, and *trustable* intelligent systems—although still with a long way to go.

In this context, it is increasingly recognised that *symbolic* approaches to machine intelligence may have

1724-8035/20/$27.50 © 2020 – IOS Press and the authors. All rights reserved

a critical role to play in overcoming the current limitations arising from the intrinsic opacity of sub-symbolic approaches [6,7,8,9]. Symbolic and sub-symbolic approaches are somewhat *complementary* to each others: while the latter ones are inherently opaque, fuzzy, and data-eager, the former ones are human-intelligible, exact, and parsimonious in terms of data. At the same time, while symbolic approaches often require human experts to manually encode symbolic knowledge, sub-symbolic approaches typically support some form of automatic learning from data.

Accordingly, it becomes fundamental to understand whether and to what extent a combination of symbolic and sub-symbolic approaches may contribute in making AI more inspectable, interpretable, or explainable. As a first step in this direction, in this paper we assess the current state of the art of integration of symbolic and sub-symbolic techniques under the XAI perspective. We provide an overview of the main models, methods, and technologies from the literature that simultaneously *(i)* propose a combination of symbolic and sub-symbolic AI approach (or vice versa), and *(ii)* explicitly target – or have the potential to be exploited for pursuing – XAI tasks or goals.

We categorise the surveyed works according to an original taxonomy discriminating contributions depending on *how* the integration of symbolic and sub-symbolic approaches is realised. For each approach we provide a brief description, a technological assessment of the software tools possibly available, and we discuss how the synergic exploitation of both symbolic and sub-symbolic techniques can bring benefits w.r.t. the aforementioned properties, as well as the kind of explanation the approach can provide.

Our assessment aims at *(i)* providing a way to compare, analyse, and evaluate the different capabilities of existing models and techniques w.r.t. XAI; *(ii)* identifying possible directions to advance the state of the art of AI, as far as the integration of symbolic and sub-symbolic approaches to automatic reasoning and machine learning is concerned; *(iii)* providing designers and developers with methods, techniques, and tools for knowledge extraction towards transfer learning; *(iv)* identifying a road map for principles and applications of explainable AI.

Accordingly, the paper is organised as follow. Section 2 provides some context information along with some definitions upon which the classification of the works is based. Section 3 presents and organises the main symbolic and sub-symbolic techniques for XAI devised out according our taxonomy. Section 4 collects and displays information from the selected approaches in form of tables, graphs, and word clouds. Finally, Section 5 provides for an overall view on symbolic and sub-symbolic techniques for XAI, offering some interpretation criteria on their current state of the art and perspectives; conclusive remarks are reported as well.

2. Context and definition

Explainable AI is an emerging field, focussing on the design and development of intelligent systems that allow humans to understand the reasons behind their recommendations or decisions, making it possible to know the data, the rationale, and the arguments leading to a particular outcome, other than to question or correct them[1].

The XAI research area has gained momentum in recent years, probably due to the occurrence of a number of related phenomena, including, but not limited to: *(i)* the pervasive adoption of data-driven AI which characterised the last decade, with associated growing expectations; *(ii)* the consequently-increased sensitivity of the world's public opinion w.r.t. data-management issues; *(iii)* the appearance of data-related regulations in most developed countries—as the European Union's GDPR[2]. However, many contributions in the XAI field are rooted on works published in the early 90s, as witnessed by some of the contributions surveyed in this paper. It is indeed well recognised that not all approaches to machine intelligence are equally understandable and interpretable in the eyes of human observers. In particular, the opacity of most techniques exploited by modern sub-symbolic AI – like neural networks – is a well-understood issue, in the same way as the interpretability of other approaches laying at the edge between the symbolic and sub-symbolic worlds—like decision trees, or rules.

In order to avoid confusion on the meaning of terms, in this section we provide a definition for fundamental notions – namely, interpretation, and explanation –, as well as the classification of the kind of explanation we can find in the surveyed techniques and the purposes for which an explanation is sought. Notions and distinctions are useful for understanding the discussion of models and their classification in the next section.

[1] https://www.ai4eu.eu
[2] https://gdpr-info.eu

2.1. Interpretation vs explanation

Interpretability and *explainability* are desirable properties for intelligent systems. We may briefly and informally define XAI as the corpus of literature and methods aimed at making sub-symbolic AI either more interpretable for humans, possibly by automating the production of explanations. However, despite the many definition attempts, there is still no common agreement on a shared, unambiguous definition for the two terms.

Based on the preliminary work by Ciatto et al. [10], and by drawing inspiration from computational logic, we let the term *interpretation* indicate the subjective relation that associates each representation with a specific meaning in the domain of the problem. In AI algorithms, interpretability refers to the cognitive effort required by human observers to assign a meaning to the way the algorithm works, or motivate the outcomes it produces. Indeed, in those contexts, the notion of interpretability is often coupled with properties as algorithmic transparency (characterising approaches which are *not* opaque), decomposability, or simulatability.

As far as the term *explanation* is concerned, we trace back its meaning to the Aristotelian thought, other than the Oxford dictionary definition. In particular, the Oxford dictionary defines explanation as a set of statements or accounts that makes something clear, or, alternatively, the reasons or justifications given for an action or belief. Thus, an explanation is an activity aimed at making the relevant details of an object clear or easy to understand to some observer.

Accordingly, the concepts of explainability and interpretability are basically *orthogonal*. However, since they are not unrelated notions, it may happen that an explanation could be given by exploiting the interpretability feature. This is the case of *explanation by model simplification* [11], where a poorly-interpretable model is translated into another – a more interpretable one –, having "high fidelity" [12] w.r.t. the first. The translation process of the first model into the second one can be considered as an explanation. For instance, as we further discuss in the next section, there exists a number of methods for *extracting* decision trees out of sub-symbolic classifiers. In this case, the extraction method is technically an explanation. Accordingly, we say that a system is explainable if there is a way to explain any of its opaque components.

2.2. Kind of explanation

According to the main impact surveys in the XAI area [6,12], the explainability feature of a system can be classified as either explainability *by design* or *post-hoc* explanability.

Explainability by design. Methods in this category aims at creating interpretable or explainable intelligent systems *by construction*. This category can be further decomposed into two sub-categories:

transparent box design containing methods supporting the creation of predictive models that are inherently interpretable, requiring no particular manipulation;

logics as constraint containing methods supporting the creation of predictive models – possibly including or involving some black box component – whose behaviour is constrained by a number of symbolic and intelligible rules, usually expressed in terms of (some subset of) first-order logic.

Post-hoc explanability. Methods in this category aims at making intelligent systems interpretable or explainable *ex-post*, i.e., by somehow manipulating some poorly interpretable pre-existing system. This category can be further decomposed into the following sub-categories:

text explanation where explainability is achieved by generating textual explanations that help to explain the model results; methods that generate symbols representing the model behaviour are also included in this category, as symbols represent the logic of the algorithm through appropriate semantic mapping;

visual explanation techniques that allow the visualisation of the model behaviour; several techniques existing in the literature comes along with methods for dimensionality reduction, to make visualisation human-interpretable;

local explanation where explainability is achieved by first segmenting the solution space into less complex solution subspaces relevant for the whole model, then producing their explanation;

explanation by example allows for the extraction of representative examples that capture the internal relationships and correlations found by the model;

model simplification techniques where a completely-new simplified system is built, trying to optimise similarity with the previous one while reducing complexity;

feature relevance methods focus on how a model works internally by assigning a relevance score to each of its features, thus revealing their importance for the model in the output.

2.3. Goals of explanation

XAI research so far has highlighted several goals that can be achieved by an explainable model. Once again, there is no shared definition of what XAI goals are, and, unfortunately, just a handful of contributions have attempted to define those goals by adopting a conceptual perspective. However, in the reminder of this work we stick on the following XAI goals and corresponding definition, also proposed in [6]:

trustworthiness as the confidence in the fact that a model will act as expected when facing of a particular problem;
causality as inference and discovering of causal relationships from data variables;
transferability as the property of a model of being transferable, therefore understandable and implementable; usually, in order to achieve this goal, it must be possible to recover all the constraints to which the model is subject;
informativeness as the ability of a model to provide information on the problem faced as well as on the context;
confidence as the assurance that a model is providing the correct answer;
fairness as the ability to achieve and guarantee equity in a model;
accessibility as the easiness to understand a model and use it;
interactivity as the readiness of a model to be interactive with the user;
privacy awareness as the possibility for the users to be aware of the risks and of the degree of protection.

These goals can help discriminating the many diverse purposes for which different explainability techniques for sub-symbolic approaches are developed.

2.4. Symbolic vs Sub-Symbolic AI

As discussed above, the focus of this survey is not on the whole spectrum of XAI methods, but rather on the ones laying at the intersection between symbolic and sub-symbolic AI—sometimes referred as *hybrid* in the literature. Note that not all hybrid methods are directly targeting the needs of XAI.

More precisely, elaborating on *how* the hybridisation of the model is performed actually allows us to build the taxonomy for the surveyed works. Two main hybridisation schemes are currently leading the research in the field:

integration where symbolic and sub-symbolic approaches are blended together in an unique model that includes features of both;
combination where symbolic and sub-symbolic approaches remain identifiable as separate blocks which are jointly exploited in order to produce an explainable intelligent system.

In both schemes, the adjective "symbolic" is often an *alias* for "logic-based". In fact, as it is further discussed in the next section, most symbolic approaches actually leverage on some sort of logic—there including simple if-then rules. This is unsurprising, as logic-based approaches already have a well-understood role in building intelligent declarative systems [13].

On the other side, the term "sub-symbolic" is used to refer to numerical, statistical, and distributed representation of machine learning models.

3. Symbolic and sub-symbolic integration: main approaches

The integration of symbolic and sub-symbolic approaches is a research field already active in the last decades, which has acquired even more relevance with the recent momentum gained by explainable AI systems. In the following we assess the state of the art according to the original taxonomy depicted in Fig. 1, which classifies contributions based on *how* the blending of symbolic and sub-symbolic approaches is realised—integration vs combination, see Subsection 2.4.

Works are further discriminated inside the model integration category depending on the kind of logics they leverage upon: "logic and numerical" integration vs "numerical, statistical, and logic". On the other side, the model composition category is split depending on the kind of composition: symbolic knowledge extraction vs symbolic knowledge injection. The former subcategory includes those approaches where some sort of symbolic knowledge is somehow *extracted* from sub-symbolic models – namely, rules, and tree extractors –, whereas the latter includes those approaches where some sort of symbolic knowledge is *injected* into sub-symbolic models.

For each selected approach we provide a brief description, as well as a technological assessment and an analysis under the XAI perspective.

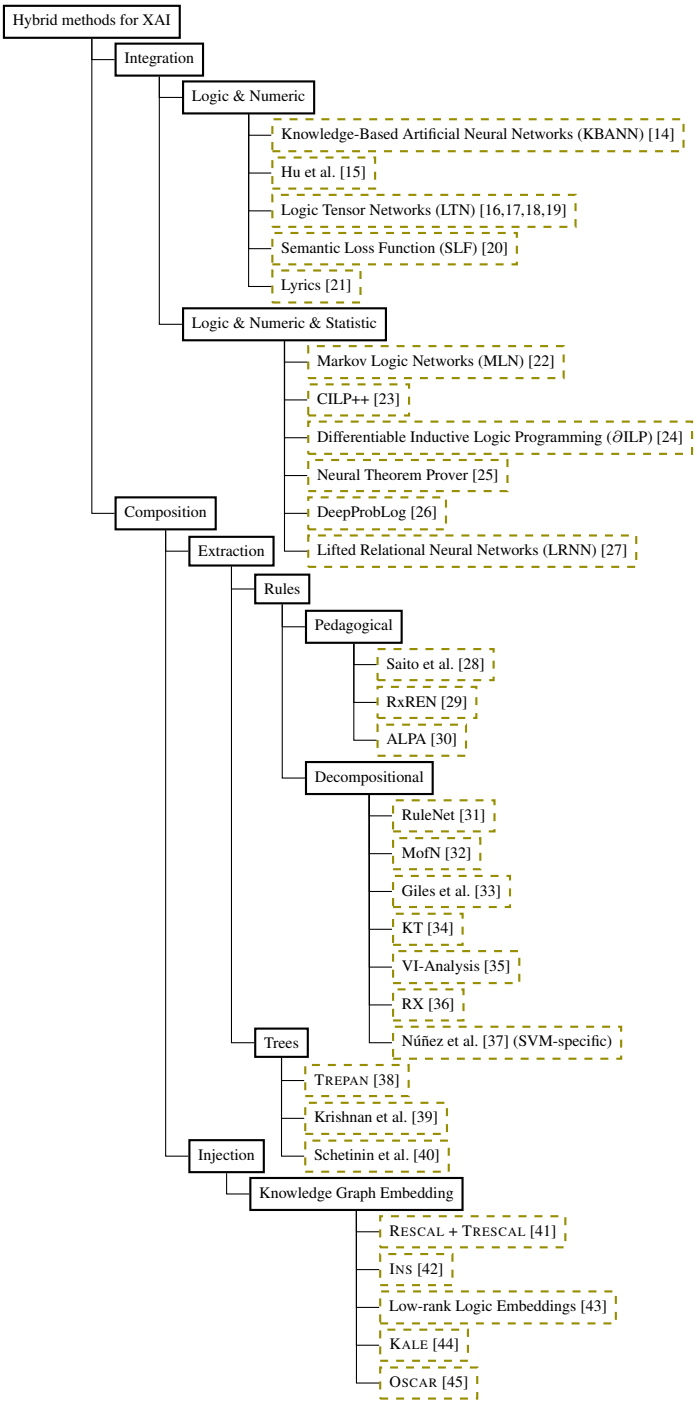

Fig. 1. Taxonomy of hybrid (symbolic + sub-symbolic) techniques for XAI.

3.1. Model integration

In this category we review the main attempts to *integrate* symbolic models (such as the logic ones) with sub-symbolic ones (such as statistical and numerical). The main research lines here are those related to the neural-symbolic computing – the study of logics and connectionism as well as statistical approaches working on the integration of computational learning and symbolic reasoning – and relational learning—focused on learning expressive logic / relational representations. Neural-symbolic integration [46] aims at building a bridge between the symbolic and sub-symbolic sides of the divide. Relational learning [47] is the union of inductive logic programming, statistical relational learning, and multi-relational data mining; it constitutes a general class of techniques and methodology for learning from structured data – such as graphs, networks, relational databases – and background knowledge.

Note that in the following we differentiate statistical and numerical approaches as follows. In statistical models, relevant aspects are either explicitly or implicitly modelled via random variables or probability distributions, and decisions are taken via Bayesian inference—e.g., graphical models, hidden Markov models, Markov networks. Conversely, in numerical models, relevant aspects are represented through one or more functions over real numbers that approximate the phenomena of interest, and decisions are taken by either minimising error or maximising the likelihood w.r.t. data—e.g., support vector machines, decision trees, random forests, or (deep) neural networks.

Accordingly, in this category we can further distinguish among approaches that integrate logic and numerical models (e.g., Logic Tensor Networks), and approaches that integrate numeric, statistical and logic models (e.g., DeepProbLog, Markov Logic Networks).

Generally speaking, the key advantage of these approaches lies in the blended integration of different models, thus allowing efficiency, on the one hand, and possibilities for interpretability and explainability, on the other.

3.1.1. Logic and numerical models integration

Here we present a number of works integrating logic and symbolic knowledge with numeric predictors such as (deep) neural network.

Integration exploits logic rules expressed via first-order logic (FOL) – or some subset of it – which are used to constrain the behaviour of one or more numeric predictors. In these cases, constraining is performed by extending the loss function used by most numeric learning algorithms – there including the back-propagation algorithm used for neural networks – with an additive, regularisation term constructed from the logic constraints. The numeric predictor is then trained "as usual", via optimisation—i.e. minimising the loss function. However, thanks to the regularisation term, the training process is more likely to select a set of parameters for the numeric predictor, which are consistent with the logic rules.

Accordingly, works in this sub-category are interesting from the XAI perspective as they support the creation of *trustworthy* hybrid systems, which provide an higher *confidence* to their users as they will behave as expected in all possible scenarios.

Knowledge-Based Artificial Neural Networks, (KBANN) 1990. KBANN [14] is one of the earliest attempts of exploiting symbolic AI to govern the structure and the behaviour of neural networks.

Differently from the other approaches described in this category, the main idea behind KBANN is not to alter the loss function used to train a neural network. Rather, KBANN is capable of devising the structure of a neural network from a symbolic knowledge base containing the user-defined, symbolic background knowledge. More precisely, KBANN assumes a stratified, Prolog-like, logic theory is available, encoding the background knowledge. Under this assumption, the KBANN algorithm aims at creating a neural network semantically reflecting the symbolic knowledge from which it was created. This step essentially sets the network structure and weights in order to reflect the rules contained into the logic theory. The resulting neural network can then be trained over data via back-propagation, in order to refine or generalise its functioning over (possibly novel) data.

According to the authors, the KBANN algorithm has proven to be useful in the area of molecular biology. In particular, neural networks attained via KBANN has been used to detect promoters in strings of nucleotides, with superior performance w.r.t. randomly initialised neural networks or other sorts of numeric predictors.

XAI perspective. KBANN is essentially a technique aimed at exploiting symbolic background knowledge to bootstrap a neural network. For this reason, we argue that KBANN provides explainability by design by exploiting logic as a constraint.

As far as XAI goals are concerned, KBANN improves the trustworthiness, confidence, and fairness of neural networks, thanks to its strong reliance on prior symbolic knowledge. However, transferability is not guaranteed in this case, as the KANN algorithm operates in a one-way fashion: apparently, there is no way to add further symbolic rules to some network built via KBANN after it has been trained on data.

Technological perspective. The KBANN algorithm is presented only in [14]: even though the authors present some experimental results, no link is provided to any sort of software implementation.

Deep Neural Networks (DNN) with Logic Rules, 2016. The recent work by [15] (no concise name is given) introduces another method for constraining a (deep) neural network behaviour via FOL rules. The proposed framework enables neural networks to be simultaneously trained on labelled data or logic rules, via an iterative distillation procedure aimed at transferring the symbolic knowledge encoded in the logic rules into the network parameters.

Unlike the main approaches described in the same category, this work is not only based on the use of logic constraints as regularisation terms. Rather, the authors propose the exploitation of two networks: a *teacher* and a *student* one. The teacher network is rule-regularised via an *ad-hoc* term added to the loss function, meaning that it is trained by keeping into account the user-provided logic rules. In particular, logic constraints are encoded into the loss function via soft logic [48]. Conversely, the student network is trained to balance between emulating the teacher network output and predicting the expected outcomes of the dataset.

In other words, this method leverages on a metaphor rooted in human education, where teachers possess the background knowledge for a given topic, and instruct students by providing them with informed suggestions helping them answer to a number of questions, until the students success rate is high enough.

According to the authors, the proposed technique has been successfully exploited in some tasks related to text analysis—namely, sentiment classification and named entity recognition. The former task consists of identifying whether a particular sentence (free text) is characterised by an underlying positive sentiment or a negative one. Conversely, the latter task consists of identifying (i.e. locating and tagging) well-know entities (characterised by a name) possibly mentioned in some portion of text, taking into account the unavoidable subtleties or ambiguities of the natural language.

XAI perspective. Similarly to other approaches in the same category, this work targets the construction of explainable intelligent systems by exploiting logic as a constraint. In particular, DNN with Logic Rules support XAI by providing trustworthiness, confidence, and fairness. Here, the distillation of logic rules increases the expectation that the student network will behave as intended, and rules may be exploited as a countermeasure biases possibly buried in the training dataset. It is worth to mention that this method represents a notable example of *transferability*. In fact, the teacher network is essentially in charge of transferring its background knowledge to the student network.

Technological perspective. An implementation based on Theano[3] is mentioned in [15], even if we were not able to find any software technology related to this work.

Logic Tensor Networks (LTN), 2017. LTN [18,19] integrate learning based on tensor networks [49] with reasoning based first-order many-valued logic [50]. The work enables a range of knowledge-based tasks using rich knowledge representation in FOL to be combined with efficient data-driven machine learning based on the manipulation of real-valued vectors.

Given data available in the form of real-valued vectors, logic soft and hard constraints, and relations that apply to certain subsets of the vectors can be specified in a compact way using FOL. Reasoning about the constraints can help improve learning, and, viceversa, learning from new data can revise the constraints thus affecting the reasoning task. The model integrates symbolic and sub-symbolic approaches and it is defined upon the Real Logic [19].

Intuitively, the logic formulæ are used to build a loss function that aims at training a network capable of approximating the truth value (in the [0,1] interval) of the formulæ given as input. This is done by searching for the best possible representation for symbolic constructs in a vector space (grounding of atoms, functions, predicates), so that the satisfiability of the network is as close as possible to 1 on the test dataset. The resulting network is able to learning from the rightly-labelled real examples, but keeps the logic imprint given in the training phase. For the sake of simplicity, the approach can be seen as placing a logic network on top of a deep neural network in order to learn the relations between abstractions, so to enable the system to self-explain.

[3] `http://deeplearning.net/software/theano`

In addition to the interesting contribution at the foundational level of AI, one of the main application of LTN has been their evaluation in a challenging yet unsolved AI task: the Semantic Image Interpretation [51]. In particular, LTN have been exploited to infer *part-of* relations between the portions of an image, to improve the effectiveness of object recognition tasks on graphical data.

XAI perspective. LTN enable the representation of relational knowledge to be infused into neural networks, in addition to the completion and distillation of knowledge through network queries. More precisely, LTN learn from numerical data and logic constraints, enabling approximate reasoning on unseen data to predict new facts which are consistent with the logic knowledge used to constrain the learning process. The predicted facts can then be used to provide inferences and reasoning mechanisms over data.

LTN are of course an example of explainability by design, where logic is exploited to constrain the sub-symbolic model. Accordingly, we argue that LTN supports XAI by providing trustworthiness, transferability, confidence, and, potentially, fairness. In fact, the symbolic knowledge (exploited as a constraint and as background knowledge) strongly reduces the risk of neural networks learning unexpected or undesired behaviours from data, thus increasing both the trustworthiness and the confidence of the system. Furthermore, by supporting the prediction of novel facts from data, LTN support transferability of knowledge. Finally, LTN may provide fairness by letting developers express constrains aimed at preventing the learning of biased or unfair behaviours from data, assuming that constraints can be expressed via FOL fourmlæ.

Technological perspective. LTN come with a technological framework based on Python 3 and TensorFlow [16], available on GitHub[4]. It consists of a full fledged library for experimenting with the LTN model. The software is actively maintained and well documented. Furthermore, it includes examples and tests.

Semantic Loss Function (SLF), 2018. Semantic Loss [20] is another attempt of bridging neural networks and symbolic constraints via loss-function manipulation, similarly to what already described for LTN. In other words, the main idea behind Semantic Loss is to constrain the training process of a neural network via some *propositional* logic formulæ which are then encoded as part the loss function exploited by the training algorithm. Such formulæ consist of boolean variables representing input and output neurons of the networks to be constrained, possibly combined via classical logic connectors.

Many differences exists, however, both at the theoretical and the technological level, between Semantic Loss and LTN (or Lyrics). In fact, Semantic Loss mostly focuses on *propositional* logic and, therefore, it does not support quantifiers or predicates within the logic formulæ used as constraints. This trait greatly simplifies both the model and its implementation. In particular the embedding schema adopted by Semantic Loss – which maps the propositional constraints in the loss functions – simply interprets neurons activation values as probabilities and accordingly computes the semantic loss via simple algebraic operations.

According to the authors, the main goals of Semantic Loss are *(i)* to improve the predictive performance of neural networks – by allowing the training process to take background knowledge into account –, and *(ii)* to support semi-supervised learning.

XAI perspective. Similarly to LTN (and Lyrics), the Semantic Loss approach targets the construction of explainable intelligent systems by exploiting a very simple logic (the propositional one) as a constraint. In particular, Semantic Loss supports XAI by providing trustworthiness, transferability, confidence, and possibly fairness. Motivations are analogous to the ones discusses above. However, we expect Semantic Loss to be less effective than the previously-discussed solutions in supporting XAI goals such as fairness, given the reduced expressibility of the logicon which it is based.

Technological perspective. The Semantic Loss technology is available on GitHub[5] and contains the code exploited by the authors for the experiments proposed in [20]. Source code consists of a few scripts targetting Python 3, coming with no documentation except for a brief description of the project. The code has not been updated since 2019. Thus, we argue that the available code is not a full fledged technology ready for general purpose usage, but rather a successful – yet concise – experiment shared with the research community.

Lyrics, 2019. At the conceptual level, the Lyrics framework is an extension of LTN, improving the way symbolic knowledge is declaratively enforced while training the sub-symbolic part of an intelligent system. Like LTN, Lyrics transforms FOL clauses into a set of constraints that are jointly optimised during learning. However, Lyrics focuses on a more declarative ap-

[4] https://github.com/logictensornetworks/logictensornetworks

[5] https://github.com/UCLA-StarAI/Semantic-Loss

proach where users can model and implement their hybrid system via an handy interface that fruitfully mixes Python and FOL.

Thanks to its declarativeness, Lyrics can combine one or more neural networks into a single computational graph. Each neural network is mapped onto a logic predicate, when necessary, while (possibly global) constrains over the outcomes of the networks are mapped into logic formulæ. The resulting computational graph is then optimised against the available data via TensorFlow.

Similarly to LTN, the Lyrics framework leverages on fuzzy logic to create a differentiable way of measuring how much the output of a sub-symbolic predictor violates the enforced logic constraints. This measure is then added as an additional loss component in the loss function to be minimised during training, therefore acting as a regulariser. In other words, Lyrics lets the background knowledge to be enforced via regularisation in sub-symbolic black-box systems.

According to the authors, the major applications of Lyrics are related to predictive model verification, semi-supervised learning with background knowledge, collective classification [52], and text chunking. While model-checking-related applications concerns XAI, and are therefore discussed in the next paragraph, semi-supervised learning can be briefly defined as the task of learning from examples where only a subset of the examples have been labelled. Collective classification, in turn, is the task of classifying data based not only on their features but also on contextual data—such as correlations or functional relations among classes. Finally, text chunking is the task of semantically recognising sentences in textual data by tagging each word with its logic role in the sentence (noun, verb, adjective, etc). In all the discussed tasks, the performance achieved by the Lyrics technology deserve to be highlighted, as they are related to the fruitful combination of symbolic and sub-symbolic techniques.

XAI perspective. Similarly to LTN, the Lyrics approach is another means to build explainable intelligent systems by construction. In particular, it falls under the "logic as constraint" sub-category. Arguably, Lyrics supports XAI by providing trustworthiness, transferability, confidence, fairness, and accessibility. Motivations are analogous to the LTN framework, except for accessibility which is a peculiarity of Lyrics due to its declarativeness.

Technological perspective. The Lyrics framework is implemented in TensorFlow. In practice, it consists of a few Python scripts targetting Python 2—which is no longer supported by the Python Software Foundation since January 2020. The code is stable and it is available on GitHub[6]. It comes with a number of usage examples, but no actual documentation is provided apart from some brief description of the project. Furthermore, it seems that the code has not been updated since 2018. For all these reasons, we classify Lyrics as legacy software.

3.1.2. Logic and numerical & statistical models integration

Here we present a number of works integrating logic and symbolic knowledge with sub-symbolic black boxes, via statistic- or probability-based approaches.

Integrating logics and probability has a long story in AI and machine learning: the integration allows to express complexity and uncertainty of complex systems—thanks to the probability component. Moreover, complex models can be learned efficiently from examples and powerful inference algorithms can be used to answer queries about the world, or verify and improve the pre-existing symbolic knowledge. Following this purpose, most methods presented in this category exploits some form of inductive logic programming or (statistical) relational learning playgrounds, fruitfully integrated with some sub-symbolic component. The way this integration is performed, however, is what characterises each surveyed technique.

Markov Logic Networks (MLN), 2006. MLN [22] is an approach for blending FOL and probabilistic graphical models (a.k.a. Markov networks) in a single model. MLN is a first-order knowledge base with a weight attached to each formula (or clause). Together with a set of constants representing objects in the domain, it specifies a ground Markov network containing one feature for each possible grounding of each clause in the knowledge base, and the corresponding weight. Weights are efficiently learned from (pre-existing) relational databases by iteratively optimising a pseudo-likelihood measure via the L-BFGS algorithm.

A MLN can be queried by users, and answers are provided via probabilistic inference. In particular, inference in MLN is performed via the Markov chain Monte Carlo (MCMC) algorithm over the minimal subset of the ground network required for answering the query.

[6] https://github.com/GiuseppeMarra/lyrics

It is worth to be mentioned that, despite being a foundational framework, MLN are at the base of many approaches adopted towards explainable AI [5,53,54].
XAI perspective. The weighted rules – representing the underlying model of this category – are exploited by a meta-interpreter to generate an explanation in the form of a proof for a why-question. Indeed, Markov Logic supports a special type of deductive inference for MLN, known as maximum a-posteriori inference, which can be exploited to find the most probable world given some evidence. Thanks to the logic formulæ exploited to describe dependencies and risk in the network the most probable explanation can be provided following the step of the inference process. Moreover, extension to abductive reasoning can be exploited, as in [55,56,57], to find an explanation for a given observation in the light of some background knowledge: the explanation refers to the root cause for a given behaviour, and the background knowledge refers to the dependency graph of the network.

For all these reasons, we classify MLN as a technique explainable by design—since there is no black-box component. It is particularly suitable to design transparent boxes out of interpretable blocks such as relational databases and graphical models. Accordingly, we argue that intelligent systems based on MLN may be exploited to pursue XAI goals such as causality, informativeness, and interactivity as they help users in understating the causes of contextual situations via interactive *why* and *what-if* queries.

Technological perspective. The authors proposes a quantitative assessment of MLN based on the Alchemy framework[7]. Alchemy is a software package providing a series of algorithms for statistical relational learning and probabilistic logic inference, based on Markov theory. The Alchemy homepage and software distribution are not updated since 2013, and we were not able to access the code, thus we are not able to provide any further technological assessment.

CILP++, 2014. CILP++ [23] is a model aimed at performing inductive logic programming (ILP) via bottom clause propositionalisation and neural networks.

Similarly to relational learning, ILP aims at inducing novel rules out of example data. However, in ILP, example data consist of logic clauses, describing *(i)* both positive and negative examples of the to-be-learned rule, and *(ii)* the background knowledge, containing a number of rules the to-be-inducted ones may exploit in their bodies.

CILP++ leverages on *(i)* neural networks to make ILP faster, and on *(ii)* propositionalisation to make the construction of neural networks out of arbitrary logic theories possible. More precisely, propositionalisation is a preliminary step, which is necessary to convert the example clauses into real vectors and the background knowledge into a multi-layered neural network to be fed with those vectors. Of course, the structure of this network reflects the rules contained in the background knowledge, and the input layer contains a neuron for each possible atom used in the background knowledge.

The resulting network is then trained against the positive and negative examples. In this way, the weights of the network are optimised to select the atoms of the to-be-induced rule—which are thus encoded in the structure of the trained network, in a sub-symbolic form. The extraction rule proposed in [58] is finally used to *extract* the induced rule from the neural network, bringing it back in symbolic form.
XAI perspective. CILP++, like other works focussing on logic induction or relational learning, can be conceived as explainable by construction exploiting logics as a constraint technique—despite it also exploits rule-extraction techniques similar to the ones used for "model simplification", which we describe later in this section. We argue that CILP++ can be exploited to pursue XAI goals such as transferability, informativeness, fairness, and accessibility. Transferability derives from the possibility of transferring both the source and the induced logic clauses from one two different instantiations of CILP++, possibly targetting similar problems. Informativeness and fairness derive from the exploitation of the induced information as a means to revel hidden information of biases, possibly buried in data. Accessibility derives from the declarativeness of NTP, where the potential of sub-symbolic AI is encapsulated behind a logic-based front end.

Technological perspective. The CILP++ technology consist of a C++ project hosted by SourceForge[8], and, more recently, by GitHub[9]. We only analysed the GitHub-hosted one, as it is the most recent one. The project consists on bare, undocumented C++ sources, coming with no release, instruction, or build automation support. The codebase has not been updates since 2017.

Neural Theorem Prover (NTP), 2017. NTP [25] are neural networks acting as logic reasoners (a.k.a. theorem provers). They are built by taking inspiration from backward-chaining-based reasoning algorithms, as in Prolog. In particular, the neural network is recursively

[7] http://alchemy.cs.washington.edu

constructed to encapsulate the knowledge encoded in some logic theory, and trained to correctly answer to all possible queries on such theory. Of course, the structure of the resulting network reflects the structure of the clauses contained into the source logic theory. However, differently from the other techniques presented in this sub-category, both theories and queries supported by NTP can contain logic variables, as NTP is able to calculate, at the neural network level – i.e., in the sub-symbolic model –, the logic unification. In other words, NTP perform symbolic reasoning on top of sub-symbolic and distributed representations of knowledge.

Thanks to their peculiarities, NTP overcome a number of limitation characterising similar approaches such as LTN and CILP++. For instance, logic theories require no preliminary grounding of clauses to be compatible with NTP. Furthermore, NTP support multi-hop reasoning.

Moreover, by exploiting the distributed representation of knowledge, the NTP tend to place the vector representations of similar symbols in the close proximity of their corresponding vectorial space. For this reason, NTP can induce novel rules to be added to the symbolic theory, given prior assumptions about the admissible structures of the logic relationships to be induced. Thus, NTP can also be considered as a tool for performing inductive logic programming (and related relational learning).

According to the authors, NTP can be exploited to perform link prediction in knowledge bases. Big handcrafted logic theories may contain some holes, in terms of missing facts or rules, and NTP may help in filling them by inducing the missing information.

XAI perspective. From the XAI point of view, we classify NTP as a means to reach explainability by construction, by using logics as a constraint. Similarly to CILP++, NTP can be exploited to pursue XAI goals such as transferability, informativeness, fairness, and accessibility, for similar reasons.

Technological perspective. Despite some experiments on NTP (and their results) are mentioned in [25], no link is provided to any sort of software artefact, and we were not able to find any software resource on this topic.

Differentiable Inductive Logic Programming (∂ILP), 2017. ∂ILP [24] is another means for ILP leveraging on neural networks. As an hybrid ILP system, it aims to combine the advantages classical symbolic induction (e.g., data-efficiency, comprehensibility, transferability, and generalisation capabilities) with the advantages of the sub-symbolic systems (e.g., robustness to noisy or ambiguous data in training).

The main idea behind ∂ILP is to mimic logic deduction on definite clauses via a neural network. However, differently from NTP, ∂ILP perform deduction using forward chaining, instead of backward chaining.

Briefly speaking, the authors re-interpret ILP as a binary-classification problem. As for other similar approaches discussed in this category, a neural network is constructed in such a way that its structure reflects a grounded version of the background knowledge. The resulting network is then trained to minimise the cross-entropy with respect to positive and negative examples.

∂ILP is essentially a foundational work. However, a number of experiments are discussed in [24] showing how ∂ILP outperforms previous ILP system in term of the number of tasks it can induce.

XAI perspective. The classification of ∂ILP under the XAI perspective is quite straightforward. Similarly to other hybrid approaches to ILP or relational learning, we classify ∂ILP as explainable by construction, exploiting logics as a constraint technique. Similarly to CILP++, and NTP, ∂ILP can be exploited to pursue XAI goals such as transferability, informativeness, fairness, and accessibility, for similar reasons.

Technological perspective. Even though some experiments on ∂ILP (and their results) are mentioned in [24], no link is provided to any sort of software artefact, and we were not able to find any software resource on this topic.

DeepProbLog, 2018. DeepProbLog [26] is another attempt of blending neural networks with logic programming, and in particular *probabilistic* logic programming.

Probabilistic logic programming is an extension of logic programming where facts and rules are enriched with probabilities, and queries are solved by not only stating if they are satisfied or not, but also to what extent—i.e., solution include probability values. As Prolog is the main language used in logic programming, ProbLog[10] is the main language used in probabilistic logic programming.

DeepProbLog is an extension of ProbLog exploiting neural networks for *(i)* computing the probabilities of facts, and *(ii)* letting neural classifiers be used as logic predicates—defined as "neural predicates" by the authors. In particular, each DeepProbLog program is translated into a tensorial computational graph – possibly including one ore more neural classifiers as

[10] https://dtai.cs.kuleuven.be/problog

sub-graphs – to be optimised via gradient descend. The structure of the computational graph reflects the structure of the rules contained into the DeepProbLog program. The optimisation step is aimed at simultaneously setting all the possible parameters regulating the behaviour of the computational graph, including the probabilities of facts and the internal weights of neural predicates. The resulting sub-symbolic system is then exploited to draw probabilistic inferences. In other words, hybrid systems based on DeepProbLog fruitfully combine probabilistic reasoning and sub-symbolic classification in a single, unified, coherent framework.

It is worth to mention that, unlike NTP, DeepProbLog requires the source logic theories to be grounded before any network can be constructed. This is why we consider DeepProbLog as a less-general approach than NTP.

According to the authors, DeepProbLog can also be used to induce new rules for a given probabilistic logic theory, exploiting its sub-symbolic representation.

XAI perspective. From the XAI point of view, DeepProbLog can be conceived as a means to reach explainability by construction by using the (probabilistic) logics as a constraint. Arguably, DeepProbLog can be exploited to pursue XAI goals such as transferability, informativeness, fairness, and accessibility. Motivations are analogous to the NTP, and CILP++ cases.

Technological perspective. DeepProbLog is available on BitBucket along with some open-source software[11]. It consists of a well-organised library targetting Python 3, ProbLog, and PyTorch[12], coming with some basic documentation. The code based was lastly update in 2019.

Lifted Relational Neural Networks (LRNN), 2018. LRNN [27] aim at performing relational learning from data via neural networks.

Similarly to DeepProbLog, LRNN exploit sets of weighted first-order formulæ as structural templates for building a neural network to be trained over the available data. The resulting network is exploited to infer latent rules buried in data and to estimate the weights of the existing clauses. Like DeepProbLog, and unlike NTP, LRNN requires the logic knowledge base to be grounded before the network construction.

According to the authors, LRNN has been successfully exploited in the context of molecular biology.

XAI perspective. From the XAI point of view, LRNN, too, can be conceived as a means to reach explainability by construction by using the (probabilistic) logics as a constraint. Again, similarly to NTP and DeepProbLog, LRNN can be exploited to pursue XAI goals such as transferability, informativeness, fairness, and accessibility, for similar reasons.

Technological perspective. Even though some experiments on LRNN (and their results) are mentioned in [27], no link is provided to any sort of software artefact, and we were not able to find any available software resource.

3.2. Model composition

In this category we review the main attempts to *combine* symbolic models – such as rule sets, decision threes, or ontologies – with sub-symbolic ones. Among the surveyed works, we identify two major research lines. Thus, we split our categorisation and our discussion in as many sub-categories.

The first sub-category deals with the translation in symbolic terms of the sub-symbolic knowledge that most numeric predictors attain from data. A number of works from the past three decades converge in this line of research. Consequently, a plethora of names have been proposed during the years for this sub-category. To avoid any confusion or bias, we stick to the name "symbolic knowledge extraction".

The second sub-category includes works and methods from the literature enabling sub-symbolic predictors to accept symbolic information as input. In fact, while most symbolic approaches can represents data structure of variable size, most sub-symbolic predictors can only work with vectorial and fixed-sized data. However, as all works in the second category focus on how to inject ontologies (a.k.a. knowledge graphs) into sub-symbolic predictors, we refer to this sub-category as "knowledge graph injection".

3.2.1. Symbolic knowledge extraction

Here we discuss the main approaches for extracting symbolic knowledge out of sub-symbolic predictors.

The main underlying assumption behind most works in this category is that, once a sub-symbolic system has been trained over large amounts of data reaching some good predictive performance, then it must have attained a distributed representation of the knowledge contained in the data. Even though unintelligible to human beings, the distributed representation is still somehow buried in the internals of that sub-symbolic systems. Assuming this is the case, then a knowledge extraction technique is a means for making the distributed representation explicit and intelligible.

It is worth to mention that the idea of extracting decision rules or trees from sub-symbolic predictors is

not new: it has been introduced several times, in many forms, and with different names and methods, since the late 80s. In fact, generally speaking, systems supporting symbolic knowledge extraction have a number of appealing features. In particular, they support a full exploitation of sub-symbolic techniques, which are the best choice when information must be mined from large amounts of data, and are usually better suited in terms of precision, robustness, and predictive performance. However, thanks to the knowledge extracted, those systems retain desirable XAI-related properties which would otherwise be lost.

Knowledge extraction techniques can be described and discriminated according to a number of orthogonal dimensions, including:

1. the structure of the symbolic knowledge they extract (e.g., decision rules, decision trees, etc)
2. the type of constraints they exploit for decision-making (e.g., linear constraints, M-of-N rules, etc)
3. the sort of sub-symbolic predictor(s) they can deal with (e.g., neural networks, support vector machines, etc)

In the reminder of this section we partition the surveyed works according to dimensions 1, and 3, then for each approach we discuss the sort of the constraints exploited. In particular, in the same way as other impactful surveys on the topic [12,59], we distinguish between techniques extracting *rule* lists and techniques extracting decision *trees*. Then, we further distinguish between pedagogical and decompositional approaches. In doing so we borrow the terminology from [59], where *pedagogical* techniques are those capable of extracting symbolic knowledge from any sort of sub-symbolic predictor – as they do *not* exploit any internal detail of the predictor under study to perform the extraction –, whereas *decompositional* techniques are those only capable of extracting symbolic knowledge from some particular sort of sub-symbolic predictor (e.g., neural networks, in most cases)—as they perform the extraction by looking at the internals of the predictor at hand.

Rules extraction. Here we focus on methods for extracting *flat* list of rules in the form

$$\begin{array}{c}\textbf{if } condition_1 \textbf{ then } outcome_1\\ \textbf{else if } condition_2 \textbf{ then } outcome_2\\ \vdots\\ \textbf{else } outcome_n\end{array}$$

out of sub-symbolic predictors, where each $condition$ is can be a conjunction or disjunction of *(i)* boolean predicates, *(ii)* linear constraints, or *(iii)* M-of-N rules over the attributes of the data used to train the sub-symbolic predictor.

We categorise the surveyed techniques for rule extraction depending on whether they are decompositional or pedagogical; then we provide some details for each technique; finally, we analyse them from the XAI perspective in an aggregate manner, given the huge similarity characterising the surveyed techniques from the XAI perspective.

Pedagogical approaches. We identified three main pedagogical approaches for rule extraction:

- the method from Saito et al. (1988) [28]
- RxREN (2012) [29]
- ALPA (2015) [30]

In particular, [28] extracts M-of-N rules out of any black-box classifier, regardless of whether it is a neural network or not. Apparently, however, this method does not support regression, and it only supports categorical attributes as conditions in the extracted rules. In spite of its limitations, this work has been proven to be effective in expert systems for diagnosis support.

On the contrary, [29] an [30] extract if-then-else rules out of arbitrary classifiers. In particular, RxREN supports datasets with mixed mode attributes (i.e., either categorical or numerical). The algorithm is based on a reverse-engineering algorithm that essentially discards insignificant attributes and discovers the variation range of input attribute for each possible outcome of the classification. For this reason, the rules extracted by RxREN are composed by linear constraints. Conversely, the ALPA rule extraction technique is the first that is applicable to any black-box model with no limitations on the nature of constraints.

It is worth remarking that pedagogical approaches are not based on the structure of the network, therefore they also work with other sub-symbolic models—even though oldest papers tend to mention neural networks more than other sorts of predictors.

Finally, it is worth to be mentioned that pedagogical extraction algorithms can essentially be described as oracle-based algorithms. In fact, in most cases the extraction algorithm works by querying the black box (which is therefore considered as an oracle), and by using the corresponding responses to build the rule list. This behaviour is repeated until the set of rules given by the white-box model converges to that of the black box. In other words, the extraction procedure termi-

nates when the rule set as whole reaches an high *fidelity* w.r.t. the original black box.

Decompositional approaches. We identify some main decompositional approaches for rule extraction:

- RuleNet (1992) [31]
- MofN (1992) [32]
- the method from Giles et al. (1993) [33]
- KT (1994) [34]
- VI-Analysis (1995) [35]
- RX (1997) [36]
- the method from Núñez et al. (2008) [37]

Generally speaking, most approaches here explicitly target a particular sort of sub-symbolic predictor. In particular, all approaches except [37] target neural networks, whereas [37] target support vector machines (SVM).

Some approaches [31,32,33] exploit some strict assumptions that limit the kind of neural networks they can manage, thus reducing their generality. For instance, the RuleNet technique described in [31] can only handle neural networks aimed at computing endomorphisms on n-sized strings of characters, and it aims at making explicit the condition-action rules exploited by such sorts of networks. At the same time, the MofN technique [32] can only handle neural networks attained via the KBANN algorithm described above. As suggested by its name, this method extracts M-of-N-like rules. Finally, the method proposed in [33] focuses on neural networks trained to act as recognisers for regular languages, and it is capable of extracting rules in the form of finite state automata for parsing these languages.

Other approaches – e.g., [34,35,36] – target general purpose neural networks. Briefly speaking, they compile networks into sets of rules with equivalent structure. There, each processing unit (neuron) is mapped into a separate rule – or a small set of rules –, and the in-going connections are interpreted as preconditions to that rule. The particular shape of preconditions – e.g., linear constraints, M-of-N constraints, etc. –, is then inferred by taking into account the weights of a neuron in-going connections, and its activation function. For instance, the KT algorithm [34] is capable of learning if-then-else rules with linear constraints out of general neural-network classifier. Similarly, the VI-Analysis [35] and RX [36] algorithms perform the same task via different procedures.

Finally, a different and noteworthy approach is described in [37]. There, the authors propose a method for extracting if-then-else rules with linear constraints out of SVM classifiers.

XAI perspective. Generally speaking, rule extraction techniques provide *post-hoc* explainability via model simplification. In fact, all the surveyed extraction procedures aims at creating a list of rule having an high-fidelity w.r.t. the source black-box predictor. This rule list can then be considered a symbolic, intelligible explanation of the source predictor. Accordingly, we argue that all these techniques may contribute to the pursuit of XAI goals as: trustworthiness, causality, transferability, informativeness, confidence, and possibly fairness. In fact, by making the inner function a black-box predictor explicit and intelligible, these techniques may increase the trustworthiness and confidence of intelligent systems. Furthermore, by providing an overview of the all the possible context-decision situation an intelligent system may face, and by making it possible to inspect which particular rule lead to a particular decision, rule extraction techniques may provide informativeness and causality. Moreover, the symbolic knowledge extracted can be translated into several forms, possibly making it compliant with symbolic intelligent systems. This of course provides for transferability. Finally, rule extraction techniques may help with fairness as well, by highlighting the biases possibly learned by sub-symbolic predictors.

It is worth to mention, however, that rule extraction technique are not the silver bullet of XAI. Issues related to accuracy, fidelity, and consistency, may easily arise in this kind of approaches, because the extracted rule list may not perfectly reflect insights of the original one. We argue that this is essentially unavoidable: the extracted rule lists are essentially approximated models, which are attained by removing (i.e. loosing) information from the source black-box. Moreover, interpretability of the extracted rule list may easily deteriorate as the amount of rules (or the amount of predicates per rule) increases—a situation which may easily arise as the complexity or the dimensionality of the black-box become non-trivial. Finally, it is worth to be noted that virtually all rule extraction techniques only focus on black-box predictors acting as classifiers. Not so much attention has been devoted by the academic community to the extraction of rules out of sub-symbolic regressors, as well as black boxes aimed at performing unsupervised learning tasks.

As a side note concerning SVM-based rule extraction techniques, it is worth to be mentioned that, although they have been known to produce classifiers that are easily comprehensible, they often approximate

secondary models of worse accuracy [60]. Moreover, even though these models may be reasonably understandable from an expert perspective, they still lack the simplicity and familiarity to an individual user that often intelligent systems have to provide, as in the case of recommendation.

Technological perspective. In spite of the many surveyed papers for rule extraction algorithm, only one of them comes with some actual implementation: ALPA. The implementation of ALPA is available for download on the web page of the Applied Data Mining Research Group at the University of Antwerp[13]. It consists of a plugin for the Weka data-mining framework[14]. Thus, it is JVM-based software coming with a detailed manual. Apparently, the ALPA source and binary code was published in 2013 and never been updated since then.

Decision trees extraction. Here we focus on methods for extracting *hierarchical* decision tree out of sub-symbolic predictors.

Generally speaking, extracted decision trees are ordinary decision trees whose nodes are represented by rules consisting of a conjunction or disjunction of *(i)* boolean predicates, *(ii)* linear constraints, or *(iii)* M-of-N rules over the attributes of the data used to train the sub-symbolic predictor, similarly to the aforementioned decision rules. In other words, the main difference with decision rules lays in the hierarchical nature of decision choices.

Given the small amount of techniques for decision tree extraction surveyed in this section, we do not split our discussion any further to distinguish between pedagogical or decompositional approaches. Rather we provide this information as part of the details description of each method, provided below. We provide a joint discussion of decision tree extraction methods from the XAI perspective, at the end of the paragraph.
Surveyed methods. We identify three main approaches for decision tree extraction:

- TREPAN (1996, pedagogical) [38]
- the method by Krishnan et al. (1999, pedagogical) [39]
- the method by Schetinin et al. (2007, decompositional: random-forest-specific) [40]

TREPAN is a pedagogical tree extraction algorithm that extracts decision trees from sub-symbolic models. TREPAN grows a tree through recursive partitioning, using a best-first expansion strategy, towards M-of-N-like, tree-structured rules. The black box model – be it a neural network, a support vector machine, or any other model that can be used for classification – is used as an oracle to answer questions of class belongingness on artificially-generated data points. It also exploits the active learning process to additionally generate data points according to network constraints.

Along the same line, [39] proposes decision tree extraction from neural networks. Unlike TREPAN, however, the internal structure of the neural network is taken into consideration in the process of decision tree construction. Furthermore, while TREPAN leverages on a restricted form of active learning, the method proposed by [39] leverages on a genetic algorithm. Finally, it is worth mentioning that the latter algorithm supports the extraction of trees of a given size. In other words, the size of extracted tree can be tuned.

Finally, [40] proposes an approach for the probabilistic interpretation of Bayesian decision trees ensembles (a.k.a. random forests) as a single decision tree. Classification confidence for each tree in the forest is calculated by exploiting training data: the decision tree covering the maximum number of correct training examples is selected, keeping the amount of classification errors in the remaining examples minimal. Unlike the previous ones, this method of explanation does not extend the input data set with random additional data and cannot be generalised to other types of sub-symbolic black boxes.
XAI perspective. From the point of view of XAI, decision tree extraction methods are quite similar to rule extraction ones, thus similar concerns fit their case. Accordingly, we argue that decision tree extraction techniques provide *post-hoc* explainability via model simplification, and help in pursuing XAI goals such as trustworthiness, causality, transferability, informativeness, confidence, and fairness.

In spite of the many similarities with rule extraction techniques, a remarkable peculiarity of decision trees extractors is worth to be mentioned: as hierarchical models, they are less prone to interpretability issues when the complexity or dimensionality of the source predictor grows.

Other critical aspects remain however unresolved w.r.t. rule extraction techniques. These include issues arising from the potential lack of fidelity, as well as the concentration of practically all the decision tree extraction techniques on the sole case of classification tasks—leaving others kinds of tasks in machine learning essentially uncovered.

Technological perspective. As far as implementations of decision tree extraction facilities are concerned, we were only able to find an ancient C-based implementation of TREPAN on the author's homepage. The archive only contains the C source code of TREPAN plus some basic instruction and a `Makefile` for build automation. We classify this technology as a legacy experimental project.

3.2.2. Knowledge graph injection

Inspired by the benefits of logic background knowledge that can lead to (more) interpretable system, knowledge injection is a field of research in some way specular to knowledge extraction. It deals with the injection of symbolic knowledge into sub-symbolic models.

Even though knowledge injection is in some sense already provided by many works described in Subsection 3.1, where symbolic and sub-symbolic techniques are integrated in an unique model – for instance, by using logics a constraint to inject background knowledge into neural systems [51] –, in this category we focus on works performing a particular sort of knowledge injection—namely, knowledge graph embedding. The key idea is to embed components of an ontology (a.k.a. knowledge graph, or simply KG) – including entities and relations – into continuous vector spaces, to allow neural networks to accept such a type of structured information as input and take advantage of its background knowledge to perform ordinary machine learning tasks.

Most of the currently-available techniques perform the embedding task only on the basis of observed facts. Given a KG, knowledge graph injection techniques first represent entities and relations in a continuous vector space, and then measure facts plausibility exploiting some scoring function. Entity and relation embeddings can be obtained by maximising the total plausibility of observed facts.

During this whole procedure, the learned embeddings are only required to be compatible within each individual fact, and hence might not be predictive enough for downstream tasks [41,42]. As a result, more and more researchers have started to add other types of information, including logic rules [41,43,44], in order to learn more predictive embeddings.

The noteworthy approaches that we deem significant for the purpose of this survey – as they combine symbolic and sub-symbolic models – are:

- RESCAL + TRESCAL (2015) [41]
- INS (2015) [42]
- Low-rank Logic Embeddings, LLE (2015) [43]
- KALE (2016) [44]
- OSCAR (2019) [45]

In particular, [41,42] exploit rules to refine embedding models aimed at KG completion. KG completion is formulated as an integer linear programming problem, where the objective function is generated from embedding models and constraints are generated from rules. Facts inferred in this way are the most preferred by the embedding models and comply with all the rules. By incorporating rules, these approaches can greatly reduce the solution space and significantly improve the inference accuracy of embedding models. TRESCAL [41] is an extension of RESCAL, requiring the arguments of a relation to be entities of certain specified types.

Along this line, other works – e.g., [43,44] – propose approaches that embed KG facts and logic rules simultaneously in a unified framework. In particular, in INS, formulæ are injected into the embeddings of relations and entity-pairs, i.e., the embeddings are estimated such that predictions based on them conform to given logic formulæ. KALE, on the other side, represents rules as complex formulæ modelled by t-norm fuzzy logics. Embedding then amounts to minimising a global loss over both atomic and complex formulae. Thus embeddings are learnt as compatible with rules.

In [45] the authors propose a method, OSCAR, for injecting task-agnostic knowledge from a KG into a neural network during the training. OSCAR is a pre-training regularisation technique capable of injecting world knowledge and ontological relationships into a deep neural network: the expert knowledge is exploited as a regulariser for the network.

It is worth noting that in all these approaches rules are modelled separately from embedding models, serving as post-processing steps: this is why we classify these work as combination and not integration. Furthermore, all these works share a common drawback, in that they have to instantiate universally-quantified rules into ground rules before learning their models. This is called grounding procedure, and can be time- and space-inefficient—especially when dealing with big data scenarios or in case of rules complexity.

XAI perspective. From the point of view of XAI, knowledge graph embedding methods are aimed at providing explainability by design using logics as a constraint. Arguably, they help in pursuing XAI goals such as trustworthiness, informativeness, confidence, and fairness. In fact, the background knowledge injected by embedding knowledge graphs may increase

the confidence and the trustworthiness of users by reducing the risks related to sub-symbolic systems acquiring undesired or biased information from data. Furthermore, the same background knowledge may be used to contextualise decisions and suggestions provided by sub-symbolic part of an intelligent system, thus improving informativeness. Finally, background knowledge may be used to prevent or compensate biased data, thus potentially improving fairness.

Technological perspective. Among the five surveyed methods for KG injection, only two works include with some public software resource. These are INS and Low-rank Logic Embeddings. In particular, the source code of the experiments proposed in [42] by the authors of INS is available on GitHub[15]—as bare and undocumented Java code. Similarly, the source of Low-rank Logic Embeddings is available on GitHub as well[16]—as Scala sources, including instructions and build automation support. In both cases, the source code is not updated since 2015.

4. Statistics, Data Collections & Discussion

In this section we collect and discuss information from the sources and the selected approaches, and display it in form of tables and word clouds.

4.1. Word cloud

Fig. 2 shows the word cloud generated from all the papers this survey is based on. In order to give some meaning to this limited yet synthetic view of the literature, the ten most evident words in the cloud could be divided into three subcategories:

- *explanation*, *systems*, *model*
- *learning*, *neural*, *network*, *training*, *value*,
- *logic*, *rule*.

In short, the picture could be interpreted as suggesting that integrated symbolic and sub-symbolic techniques for XAI are mostly exploited to give an explanation of sub-symbolic systems exploiting the expressive power of logics in modelling knowledge into rules, formulæ and states. Neural networks are chosen as typical representatives of sub-symbolic models, while logic rules are chosen to represent the symbolic approaches.

The cloud also highlights the relatively-high frequency of other words, related to the *logic* one (representative of the symbolic approaches), such as: *knowledge*, *formula*, *variable*, *clauses*, *interpretation*, *ontology*, *fuzzy*—all of them basically reinforcing the (quite obvious) idea that the knowledge to be discovered should be represented exploiting some kind of logic rules to formalise the systems and its properties. On the other side, the cloud shows some high-frequency words related to the machine learning domain (which includes most of the sub-symbolic approaches), such as: *function*, *output*, *input*, *classification*, *space*, *features*—somehow hinting at the main features of sub-symbolic approaches, such as the efficiency in finding solutions and learning in a very large data space.

Further scrutiny reveals that terms *understanding*, *black*, *hidden*, *information*, and *trust* also appear quite often. This hints at the main XAI goals these hybrid approaches aspire to achieve.

Other interesting words emerging from the cloud are *prediction*, *reasoning*, *inference*, and *decision*, possibly reflecting one of the main purposes of the symbolic / sub-symbolic integrated approaches for XAI—namely, predicting and reasoning over the system knowledge in order to take autonomous decision, and supporting the design and development of intelligent systems as human-centred AI technologies and applications. Finally, words like *social*, *human*, and *interactions* spotlight the social component of the intelligent systems target for the surveyed hybrid approaches.

4.2. Timeline Analysis

Fig. 3 presents the papers in this survey on a timeline. At first glance, it can be seen that forerunners works – which characterised the decades between the 1990s and 2010s – are those related to knowledge extraction—in the form of both rules and trees, precisely in this order, to underline the influence of the former on the latter.

In 2015, research works on model integration debuted on the XAI playground, and soon gained momentum, reaching the high concentration of works that we see today—although a couple of (theoretical) precursors can be found from the early 90s (e.g., [14]). This underlines two basic concepts – perhaps obvious, but worth remembering – of today's AI scenario: *i)* the large amount of environmental data and the availability of increasingly-advanced technologies have made it possible to exploit integrated symbolic / sub-symbolic approaches in real applications of intelligent systems; *ii)* the need for these systems to be explainable is today more fundamental than ever.

It is also worth noting how the progress of technology and techniques has led to the exploration of approaches for the injection of knowledge, however dominated – in the same years – by approaches always

Fig. 2. The cloud of words obtained by the main keywords extracted from the papers subject of the survey.

linked to “logics as constraints”, yet based on model integration. Model integration has several advantages, discussed in the next sub-section: first of all, that of being explainable by design.

4.3. XAI Analysis

Table 1 and Table 2 summarise the XAI perspective analysis provided in Section 3 for each approach.

Kind of explanation. At a glance, Table 1 shows symbolic/sub-symbolic integrated/combined techniques for XAI are actually concentrated in just two categories w.r.t. the type of explanation—namely, “logics as constraint” and “explanation by simplification”.

About the techniques that *integrate* symbolic and sub-symbolic approaches, there is actually one item labelled as transparent box design – the Markov Logic networks – since they are the only ones without components classifiable as black boxes. Apart from this exception, other techniques that *integrate* symbolic and sub-symbolic approaches exploit the use of “logics as constraints”: the integration is indeed implemented by binding the network (via logic constraints) during the training phase in order to produce a model that comply with the background knowledge (logic constraints).

The use of knowledge in the form of logic declarations or constraints in the knowledge base has shown not only to improve explainability, but, in some approaches, also to improve performance compared to approaches based exclusively on data [19,51]. The main positive effect of these approaches is that hybrid and integrated approaches provide robustness to the learning system when errors occur in the training data labels.

Furthermore, many of these approaches have proven to be able to learn and reason jointly with both symbolic and sub-symbolic representations and inferences, sometimes enabling a probabilistic expressive logic reasoning [26].

A different perspective on XAI hybrid models is to enrich the knowledge of the black box models with symbolic knowledge, as proposed by the techniques categorised as *model composition*. In particular, this can be done by binding the neural network thanks to semantic KB and the like.

This is the case of *composition* of symbolic and sub-symbolic techniques that can be “explained by design” through the “logics as a constraint” technique. Unlike the integrated models, in this case the logic constraints (i.e. the knowledge graphs) are injected into the symbolic sub-model as input (once appropriately translated

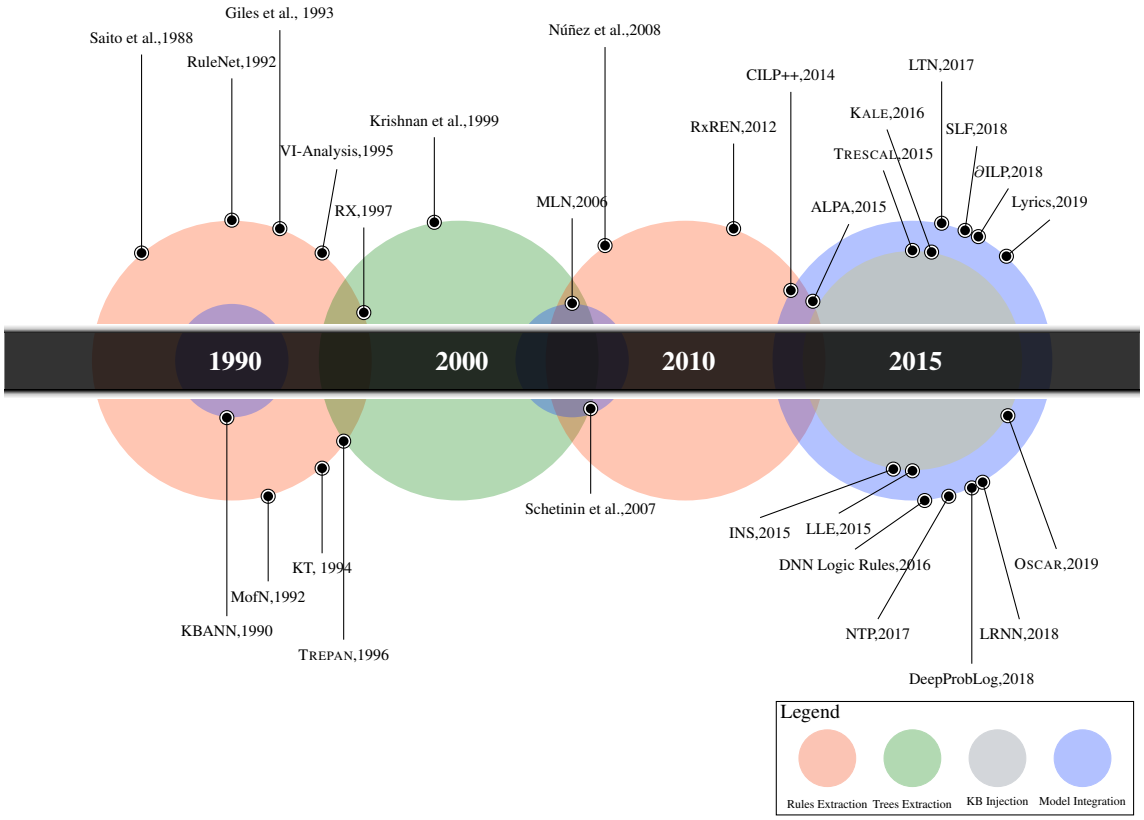

Fig. 3. Surveyed works timeline.

into numerical vectors). In this case the output of the sub-symbolic model is bounded by the constraints imposed as input.

Another hybrid approach is to map a system of black boxes onto a white box, or, a more interpretable twin. Along this line, any technique that reduces the complexity of the model or simplifies the results should be considered as a XAI approach. The size of this translation, in terms of complexity or simplicity, should correspond to how explainable the resulting model is. An underlying problem that remains unsolved is that the gain of interpretability provided by such XAI approaches may not be easy to quantify. The definition of general and shared metrics allowing XAI approaches to be evaluated remains an open challenge.

About the sort of explanation, all the approaches categorised as composition of techniques via knowledge extraction are included in the model simplification category. The approaches are those related to the extraction of rules or trees from the sub-symbolic model.

Decision trees have always been one of the most used categories of transparent models. Also based on the number of approaches collected in this survey, the literature on simplification and generation of the decision tree is very broad. The fact that a large majority of the application of these models does not fall within the field of AI (and not even in information technologies) mostly means that experts from other sectors usually feel comfortable when interpreting the results of these models. In fact, decision trees are models that perfectly achieve the goal of trustworthiness. However, they have poor scalability properties when compared to other models: therefore, they are difficult to apply in scenarios where predictive performance is of paramount importance.

Overall, Table 1 shows that obtaining a model as a transparent box is almost unlikely (due to the need of integrating the two techniques). However, the table also highlights that there are unexplored research directions that it might be worth exploring—such as textual, visual, local, by example and by feature explanations. All these directions become worthy of consideration and may encompass promising research perspectives to be taken into account. It is worth noting that

Table 1
Symbolic/sub-symbolic techniques for XAI: kind of explanation

		explainable by design		post-hoc explainability					
		transparent box design	logic as constraint	text explanation	visual explanation	local explanation	explanation by example	model simplifaction	feature relevance
model integration	KBANN		✓						
	DNN with Logic Rules		✓						
	Logic Tensor Networks		✓						
	Semantic Loss Function		✓						
	Lyrics		✓						
	Markov Logic Networks	✓							
	CILP++		✓						
	Neural Theorem Prover		✓						
	∂ILP		✓						
	DeepProbLog		✓						
	Lifted Relational NN		✓						
model composition KB extraction	Saito et al.							✓	
	RxREN							✓	
	ALPA							✓	
	Núñez et al.							✓	
	KT							✓	
	RX							✓	
	RuleNet							✓	
	Giles et al.							✓	
	MofN							✓	
	VI-Analysis							✓	
	TREPAN							✓	
	Krishnan et al.							✓	
	Schetinin et al.							✓	
model composition KB injection	RESCAL + TRESCAL		✓						
	Low-rank Logic Embeddings		✓						
	KALE		✓						
	INS		✓						
	ontology injection		✓						

textual or visual explanations are easy to obtain, once the model is simplified, from models such as decision trees: some attempts in this direction have already been made [9].

Generally speaking, methods from the "post-hoc explanation" category are usually preferred, as they do not prevent the exploitation of black-box predictors—commonly considered as better-performing in the general case. Conversely, the methods from the "explainable by design" category are best suited for critical contexts where interpretability is a major concern, and predictive performance can be sacrificed in its favour.

XAI goals. As already mentioned above, XAI definition brings together two concepts – namely, understanding and trust – that need to be addressed by the XAI model for AI. However, other purposes motivate the need for interpretable AI models—such as causality, transferability, informativeness, fairness, confidence, accessibility, interactivity, and privacy awareness. Table 2 focuses on XAI goals, based on the papers that state them explicitly as their goal. A checkmark between brackets denotes that the analysis of Section 3 has highlighted the XAI goal although it is not explicitly mentioned by the authors in their works.

Trustworthiness. Several authors agree on trustworthiness as the primary goal of an XAI model of AI. However, not all models addressing this goal explicitly talk about it in their works. According to the definition given here, trustworthiness amounts to ensure that a model will act as expected when facing a specific problem. This is why the works categorised as

Table 2
Symbolic/sub-symbolic integrated techniques for XAI: main goals

		trustworthi-ness	causality	transferability	informative--ness	confidence	fairness	accessibility	interactivity	privacy awareness
model integration	KBANN	✓			✓	✓	(✓)			
	DNN with Logic Rules	✓		✓	✓	✓				
	Logic Tensor Networks	✓		✓	✓	✓	(✓)			
	Semantic Loss Function	✓		✓	✓	✓	✓			
	Lyrics	✓		✓	✓	✓	✓	✓		
	Markov Logic Networks		✓			✓			✓	
	CILP++			✓	✓		(✓)			
	Neural Theorem Prover			✓	✓		(✓)	✓		
	∂ILP			✓	✓		(✓)	✓		
	DeepProbLog			✓	✓		(✓)	✓		
	LRNN			✓	✓		(✓)	✓		
model composition	rules extraction	✓	✓	✓	✓	✓	(✓)			
	trees extraction	✓	✓	✓	✓	✓	(✓)			
	knowledge graph injection	✓			✓	✓	(✓)			

those pursuing this goal are the ones exploiting any form of symbolic knowledge (exploited as a constraint, or as background knowledge, or as knowledge extracted / injected into a model). Indeed, this feature strongly reduces the risk of black-box model to learn unexpected or undesired behaviours from data, thus increasing both the trustworthiness and the confidence of the system. Part of the works reviewed here explicitly mention the concept of trust when stating their purpose for achieving explainability. However, as one may observe in Table 2, they represent more or less half of the contributions.

Causality. Another goal in XAI is to find the causality between the data and the variables that represent them. Several authors argue that XAI models could facilitate the task of finding relationships that could be further tested for a stronger causal link between the variables involved. A sub-symbolic model discovers only the correlations between the learning data, therefore it may not be enough to reveal the general cause-effect relationships. However, causality implies correlation, therefore a sub-symbolic model integrated with symbolic techniques could validate the results provided with inference techniques, or, provide a first intuition of possible causal relationships. Once again, Table 2 shows that causality is neither among the most important objectives nor among those mostly addressed by hybrid models. Indeed, abduction-based approaches in Markov Logic Networks fall into this category. Abduction aims at finding the root cause, by drawing a graph of the causes, thus providing an explanation for a given observation.

Transferability. The model constraints that delimit and harness the model behaviour should be easy to transfer. Explainability may also be expressed under the guise of transferability, since it can facilitate the task of clarifying the boundaries that could influence a model, thus allowing a better understanding and implementation. For this reason, an explainable model should also be easily transferable—yet again, not viceversa. According to Table 2, the number of articles that include transferability among the features that make a model explainable is quite high—indeed transferability is the second most-mentioned goal. Generally speaking, the works that integrate ILP or relational learning do not address this goal: in fact, those techniques exploit the symbolic component to induce new rules, yet they do not represent the global knowledge of the system in order to make transferability possible.

Informativeness. Intelligent systems – and therefore models that support them – are mostly used with the intention of supporting the decision-making process of agents—here intended both as software and human ones. However, the problem solved by the model is never the same as the one humans would face. Therefore, a lot of information is always required in order to be able to relate the decision that a user would take to the solution provided by the model so as to avoid misunderstandings and misleading outcomes. To this end, explainable models should provide information on the problem addressed. Most of the articles surveyed use the expression *provide information* to refer to the process of disclosing relationships between data

and the corresponding variables. All knowledge extraction techniques (regardless of whether they extract rules or trees) essentially provide post-hoc explainability via model simplification. In fact, sub-symbolic knowledge is then expressed and presented to the users via symbolic proxies – such as list or trees of rules – which can be considered and explanation of the original black box model. This is why all those techniques have *informativeness* as their main goal. Even at the level of model integration, the achievement of this goal is mentioned almost everywhere. This is certainly the most widely-used topic among the articles reviewed to support what they expect to achieve from XAI models.

Confidence. Confidence is basically mandatory in the models where reliability is expected—even more in the XAI perspective. Only half of the surveyed works deals with this feature, which seems reasonably to go hand in hand with trustworthiness. In fact, as mentioned above, some models – such as those based on ILP and RL, which are mainly used to learn latent rules and relationships – do not aim at making systems more trustworthy or confident—their (other) purposes are detailed in Section 3.

Fairness. Explainability, from a social perspective, can be considered as the ability to achieve and ensure absence of biases in the models for intelligent systems. There are strands of literature – outside the scope of this survey – that enable for an ethical analysis of the model in order to measure the level of equity achieved. Accordingly, a XAI objective should be enable the detection of possible distortion in the data. In fact, XAI could actually work as a bridge to avoid the unfair or unethical use of the model results. Among the works accounted here, almost no one explicitly mentions fairness, which instead is nowadays perceived as one of the most relevant issues. However, in Section 3 we have shown how, by increasing the degree of information, therefore of confidence and trust, many techniques already take a first step towards fairness.

Accessibility. A small portion of the surveyed contributions also aims at improving accessibility, always in order to promote explainability of a model. This is quite obvious, since the ability for humans to map their own cognitive models upon software models is surely a fundamental step towards explainability.

Interactivity. Only one contribution include the model ability to interact with the user—namely, Markov Logic Networks, which help users understanding the causes of contextual situations via interactive *why* and *what-if* queries. Yet, interaction with users should be one of the main XAI goals, so it seems likely that many future research efforts will be devoted to that.

Privacy awareness. Almost forgotten in the surveyed works, the ability to evaluate conditions relating to privacy should be accounted for by explainable models. In fact, the lack of understanding of what has been acquired, memorised, and learned by the model can lead to violations of privacy norms, thus leading to legal issues. This is a critical problem, given the main social contexts where XAI is going play a crucial role. As a result, researches in the hybrid legal/IT field look both promising and needed.

5. Conclusion

AI systems nowadays work on large amounts of data, learning from experience and making predictions with the goal of supporting human decisions or taking autonomous ones—applications range from clinical decision support to autonomous driving and predictive policing [61]. 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 *explainability* and *interpretability* in machine learning and AI has focussed on simplified models that approximate the true criteria used to make decisions.

Bridging the gap between symbolic and sub-symbolic representations is a key obstacle along the path from the present state of AI achievement towards human-level artificial intelligence—in particular in the explainable AI perspective.

Symbolic intelligence, based on the use of symbolic rules to represent, explore, and infer knowledge, has many advantages, such as naturalness, interpretability, and easiness in giving explanations. However, it has two major disadvantages: acquiring rules and learning from a huge amount of data is difficult, and scalability is poor. On the other hand, sub-symbolic intelligence has the great advantage of being able to acquire knowledge from examples, to be highly scalable, and to succeed in representing complex and highly-inaccurate knowledge bases. The benefits and drawbacks of the two approaches compensate each other: this has lead to the emergence of techniques combining and integrating them so as to get strengths and benefits from both.

In this paper we focus on explainable AI, and discuss the main techniques for the integration of sym-

bolic and sub-symbolic approaches on the general perspective of explainability of intelligent systems. On the one hand, we categorise the works that deal with model integration, where the symbolic and sub-symbolic components are no longer two separate parts of the system. On the other hand, we instead classify techniques and systems that combine the two approaches to obtain hybridisation.

Overall, in spite of the heterogeneity of the techniques and the variety of the research directions, all works demonstrate the feasibility and the potential benefits of the integration of symbolic techniques with sub-symbolic ones towards explainable AI.

Funding

This work has been partially supported by the H2020 Project "AI4EU" (G.A. 825619).

One of the authors, Roberta Calegari, has been supported by project "CompuLaw", funded by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (G.A. 833647).

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] Z.M. Fadlullah, F. Tang, B. Mao, N. Kato, O. Akashi, T. Inoue and K. Mizutani, State-of-the-Art Deep Learning: Evolving Machine Intelligence Toward Tomorrow's Intelligent Network Traffic Control Systems, *IEEE Communications Surveys Tutorials* **19**(4) (2017), 2432–2455. doi:10.1109/COMST.2017.2707140.

[4] Z.C. Lipton, The Mythos of Model Interpretability, *Queue* **16**(3) (2018), 31—57. doi:10.1145/3236386.3241340.

[5] D. Gunning and D.W. Aha, DARPA's Explainable Artificial Intelligence Program, *AI Magazine* **40**(2) (2019), 44–58.

[6] A. Barredo Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. Garcia, S. Gil-Lopez, D. Molina, R. Benjamins, R. Chatila and F. Herrera, Explainable Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI, *Information Fusion* **58**(December 2019) (2020), 82–115. doi:10.1016/j.inffus.2019.12.012.

[7] B. Goertzel, Perception Processing for General Intelligence: Bridging the Symbolic/Subsymbolic Gap, in: *Artificial General Intelligence*, J. Bach, B. Goertzel and M. Iklé, eds, Springer Berlin Heidelberg, 2012, pp. 79–88. ISBN ISBN 978-3-642-35506-6.

[8] R. Calegari, G. Ciatto, S. Mariani, E. Denti and A. Omicini, LPaaS as Micro-intelligence: Enhancing IoT with Symbolic Reasoning, *Big Data and Cognitive Computing* **2**(3) (2018). doi:10.3390/bdcc2030023.

[9] R. Calegari, G. Ciatto, J. Dellaluce and A. Omicini, Interpretable Narrative Explanation for ML Predictors with LP: A Case Study for XAI, in: *WOA 2019 – 20th Workshop "From Objects to Agents"*, F. Bergenti and S. Monica, eds, CEUR Workshop Proceedings, Vol. 2404, Sun SITE Central Europe, RWTH Aachen University, 2019, pp. 105–112. http://ceur-ws.org/Vol-2404/paper16.pdf.

[10] G. Ciatto, D. Calvaresi, M.I. Schumacher and A. Omicini, An Abstract Framework for Agent-Based Explanations in AI, in: *19th International Conference on Autonomous Agents and MultiAgent Systems*, International Foundation for Autonomous Agents and Multiagent Systems, 2020, pp. 1816–1818, Extended Abstract. ISSN 2523-5699. ISBN ISBN 978-1-4503-7518-4. http://ifaamas.org/Proceedings/aamas2020/pdfs/p1816.pdf.

[11] 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. http://dl.acm.org/citation.cfm?id=3098039.

[12] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti and D. Pedreschi, A Survey of Methods for Explaining Black Box Models, *ACM Computing Surveys* **51**(5) (2018). doi:10.1145/3236009.

[13] A. Omicini and F. Zambonelli, MAS as Complex Systems: A View on the Role of Declarative Approaches, in: *Declarative Agent Languages and Technologies*, Lecture Notes in Computer Science, Vol. 2990, Springer, 2004, pp. 1–17. ISSN 0302-9743. ISBN ISBN 978-3-540-22124-1. doi:10.1007/978-3-540-25932-9_1.

[14] G.G. Towell, J.W. Shavlik and M.O. Noordeweir, Refinement of Approximate Domain Theories by Knowledge-Based Neural Networks, in: *Proceedings of the Eighth National Conference on Artificial Intelligence*, 1990, pp. 861–866. https://www.aaai.org/Library/AAAI/1990/aaai90-129.php.

[15] Z. Hu, X. Ma, Z. Liu, E. Hovy and E. Xing, Harnessing Deep Neural Networks with Logic Rules, in: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2016, pp. 2410–2420.

[16] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard et al., Tensorflow: A system for large-scale machine learning, in: *12th USENIX Symposium on Operating Systems Design and Implementation*, 2016, pp. 265–283.

[17] W.W. Cohen, TensorLog: A Differentiable Deductive Database, *CoRR* **abs/1605.06523** (2016).

[18] L. Serafini and A.S.d. Garcez, Learning and reasoning with logic tensor networks, in: *Conference of the Italian Association for Artificial Intelligence*, Springer, 2016, pp. 334–348.

[19] L. Serafini, I. Donadello and A.d. Garcez, Learning and Reasoning in Logic Tensor Networks: Theory and Application to Semantic Image Interpretation, in: *Proceedings of the Symposium on Applied Computing*, SAC '17, ACM, New York, NY, USA, 2017, pp. 125–130. ISBN ISBN 978-1-4503-4486-9. doi:10.1145/3019612.3019642.

[20] J. Xu, Z. Zhang, T. Friedman, Y. Liang and G. Broeck, A Semantic Loss Function for Deep Learning with Symbolic Knowledge, in: *35th Conference on Machine Learning (ICML 2018)*, J. Dy and A. Krause, eds, Proceedings of Machine Learning Research, Vol. 80, PMLR, 2018, pp. 5502–5511. http://proceedings.mlr.press/v80/xu18h.html.

[21] G. Marra, F. Giannini, M. Diligenti and M. Gori, LYRICS: a General Interface Layer to Integrate AI and Deep Learning, *arXiv preprint arXiv:1903.07534* (2019).

[22] M. Richardson and P. Domingos, Markov logic networks, *Machine learning* **62**(1–2) (2006), 107–136.

[23] M.V.M. França, G. Zaverucha and A.S. d'Avila Garcez, Fast relational learning using bottom clause propositionalization with artificial neural networks, *Machine Learning* **94**(1) (2014), 81–104. doi:10.1007/s10994-013-5392-1.

[24] R. Evans and E. Grefenstette, Learning Explanatory Rules from Noisy Data, *Journal of Artificial Intelligence Research* **61** (2018), 1–64. ISBN ISBN 9780999241127. doi:10.1613/jair.5714.

[25] T. Rocktaschel and S. Riedel, End-to-end differentiable proving, in: *Advances in Neural Information Processing Systems*, 2017, pp. 3788–3800.

[26] R. Manhaeve, S. Dumancic, A. Kimmig, T. Demeester and L. De Raedt, DeepProbLog: Neural Probabilistic Logic Programming, in: *Advances in Neural Information Processing Systems 31*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi and R. Garnett, eds, Curran Associates, Inc., 2018, pp. 3749–3759.

[27] G. Sourek, V. Aschenbrenner, F. Zelezný, S. Schockaert and O. Kuzelka, Lifted Relational Neural Networks: Efficient Learning of Latent Relational Structures, *Journal of Artificial Intelligence Research* **62** (2018), 69–100. doi:10.1613/jair.1.11203.

[28] K. Saito and R. Nakano, Medical diagnostic expert system based on PDP model, in: *IEEE 1988 International Conference on Neural Networks (ICNN 1988)*, Vol. 1, 1988, pp. 255–262. doi:10.1109/ICNN.1988.23855.

[29] M.G. Augasta and T. Kathirvalavakumar, Reverse engineering the neural networks for rule extraction in classification problems, *Neural processing letters* **35**(2) (2012), 131–150. doi:10.1007/s11063-011-9207-8.

[30] E.J. de Fortuny and D. Martens, Active Learning-Based Pedagogical Rule Extraction, *IEEE Transactions on Neural Networks and Learning Systems* **26**(11) (2015), 2664–2677. doi:10.1109/TNNLS.2015.2389037.

[31] C. McMillan, M.C. Mozer and P. Smolensky, Rule induction through integrated symbolic and subsymbolic processing, in: *Advances in neural information processing systems*, 1992, pp. 969–976.

[32] G. Towell and J.W. Shavlik, Interpretation of artificial neural networks: Mapping knowledge-based neural networks into rules, in: *Advances in neural information processing systems*, 1992, pp. 977–984.

[33] C.L. Giles and C.W. Omlin, Rule refinement with recurrent neural networks, in: *IEEE International Conference on Neural Networks*, IEEE, 1993, pp. 801–806.

[34] L. Fu, Rule generation from neural networks, *IEEE Transactions on Systems, Man, and Cybernetics* **24**(8) (1994), 1114–1124.

[35] S. Thrun, Extracting rules from artificial neural networks with distributed representations, in: *Advances in neural information processing systems*, 1995, pp. 505–512.

[36] R. Setiono, Extracting Rules from Neural Networks by Pruning and Hidden-Unit Splitting, *Neural Computation* **9**(1) (1997), 205–225. doi:10.1162/neco.1997.9.1.205.

[37] H. Núñez, C. Angulo and A. Català, Rule Extraction Based on Support and Prototype Vectors, in: *Rule Extraction from Support Vector Machines*, J. Diederich, ed., Studies in Computational Intelligence, Vol. 80, Springer, 2008, pp. 109–134. doi:10.1007/978-3-540-75390-2_5.

[38] M. Craven and J.W. Shavlik, Extracting tree-structured representations of trained networks, in: *Advances in neural information processing systems*, 1996, pp. 24–30.

[39] R. Krishnan, G. Sivakumar and P. Bhattacharya, Extracting decision trees from trained neural networks, *Pattern recognition* **32**(12) (1999).

[40] V. Schetinin, J.E. Fieldsend, D. Partridge, T.J. Coats, W.J. Krzanowski, R.M. Everson, T.C. Bailey and A. Hernandez, Confident interpretation of Bayesian decision tree ensembles for clinical applications, *IEEE Transactions on Information Technology in Biomedicine* **11**(3) (2007), 312–319.

[41] Q. Wang, B. Wang and L. Guo, Knowledge base completion using embeddings and rules, in: *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.

[42] Z. Wei, J. Zhao, K. Liu, Z. Qi, Z. Sun and G. Tian, Large-scale knowledge base completion: Inferring via grounding network sampling over selected instances, in: *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, ACM, 2015, pp. 1331–1340.

[43] T. Rocktäschel, S. Singh and S. Riedel, Injecting logical background knowledge into embeddings for relation extraction, in: *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2015, pp. 1119–1129.

[44] S. Guo, Q. Wang, L. Wang, B. Wang and L. Guo, Jointly embedding knowledge graphs and logical rules, in: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2016, pp. 192–202.

[45] T.R. Goodwin and D. Demner-Fushman, Bridging the Knowledge Gap: Enhancing Question Answering with World and Domain Knowledge, *arXiv preprint arXiv:1910.07429* (2019).

[46] B. Hammer, B. Hammer and P. Hitzler, *Perspectives of Neural-Symbolic Integration*, Vol. 77, 1st edn, Springer Publishing Company, Incorporated, 2007. ISBN ISBN 354073953X.

[47] L. De Raedt, *Logical and relational learning*, Springer Science & Business Media, 2008.

[48] S.H. Bach, M. Broecheler, B. Huang and L. Getoor, Hinge-Loss Markov Random Fields and Probabilistic Soft Logic, *Journal of Machine Learning Research* **18** (2017), 109:1–109:67. http://jmlr.org/papers/v18/15-631.html.

[49] R. Socher, D. Chen, C.D. Manning and A. Ng, Reasoning with neural tensor networks for knowledge base completion, in: *Advances in neural information processing systems*, 2013, pp. 926–934.

[50] M. Bergmann, *An introduction to many-valued and fuzzy logic: semantics, algebras, and derivation systems*, Cambridge University Press, 2008.

[51] I. Donadello, L. Serafini and A.D. Garcez, Logic tensor networks for semantic image interpretation, in: *Proceedings of the 26th International Joint Conference on Artificial Intelligence*,

AAAI Press, 2017, pp. 1596–1602.

[52] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Gallagher and T. Eliassi-Rad, Collective Classification in Network Data, *AI Magazine* **29**(3) (2008), 93–106. http://www.aaai.org/ojs/index.php/aimagazine/article/view/2157.

[53] V. Belle, Logic meets Probability: Towards Explainable AI Systems for Uncertain Worlds., in: *IJCAI*, 2017, pp. 5116–5120.

[54] K. Qian, L. Popa and P. Sen, SystemER: a human-in-the-loop system for explainable entity resolution, *Proceedings of the VLDB Endowment* **12**(12) (2019), 1794–1797. doi:10.14778/3352063.3352068.

[55] R.J. Kate and R.J. Mooney, RJ: Probabilistic abduction using Markov logic networks, in: *IJCAI '09 Workshop on Plan, Activity, and Intent Recognition*, 2009.

[56] J. Blythe, J.R. Hobbs, P. Domingos, R.J. Kate and R.J. Mooney, Implementing Weighted Abduction in Markov Logic, in: *Proceedings of the Ninth International Conference on Computational Semantics*, IWCS '11, Association for Computational Linguistics, Stroudsburg, PA, USA, 2011, pp. 55–64. http://dl.acm.org/citation.cfm?id=2002669.2002676.

[57] J. Schoenfisch, J. Stulpnagel, J. Ortmann, C. Meilicke and H. Stuckenschmidt, Root Cause Analysis through Abduction in Markov Logic Networks, in: *2016 IEEE 20th International Enterprise Distributed Object Computing Conference (EDOC)*, 2016, pp. 1–8. doi:10.1109/EDOC.2016.7579386.

[58] A.S. d'Avila Garcez, K. Broda and D.M. Gabbay, Symbolic knowledge extraction from trained neural networks: A sound approach, *Artificial Intelligence* **125**(1–2) (2001), 155–207. doi:10.1016/S0004-3702(00)00077-1.

[59] R. Andrews, J. Diederich and A.B. Tickle, Survey and Critique of Techniques for Extracting Rules from Trained Artificial Neural Networks, *Knowledge-Based Systems* **8**(6) (1995), 373–389. doi:10.1016/0950-7051(96)81920-4.

[60] N. Barakat and A.P. Bradley, Rule extraction from support vector machines: A review, *Neurocomputing* **74**(1) (2010), 178–190, Artificial Brains. doi:10.1016/j.neucom.2010.02.016.

[61] R. Calegari, G. Ciatto, E. Denti and A. Omicini, Logic-based Technologies for Intelligent Systems: State of the Art and Perspectives, *Information* **11**(3) (2020), 1–29, Special Issue "10th Anniversary of Information—Emerging Research Challenges". doi:10.3390/info11030167. http://www.mdpi.com/2078-2489/11/3/167.