Back to the Feature: A Neural-Symbolic Perspective on Explainable AI

Andrea Campagner, Federico Cabitza

Back to the Feature: a Neural-Symbolic Perspective on Explainable AI

Andrea Campagner and Federico Cabitza

Universitá degli Studi di Milano-Bicocca, Milan, Italy
a.campagner@campus.unimib.it

Abstract. We discuss a perspective aimed at making black box models more eXplainable, within the eXplainable AI (XAI) strand of research. We argue that the traditional end-to-end learning approach used to train Deep Learning (DL) models does not fit the tenets and aims of XAI. Going back to the idea of hand-crafted feature engineering, we suggest a hybrid DL approach to XAI: instead of employing end-to-end learning, we suggest to use DL for the automatic detection of meaningful, hand-crafted high-level symbolic features, which are then to be used by a standard and more interpretable learning model. We exemplify this hybrid learning model in a proof of concept, based on the recently proposed Kandinsky Patterns benchmark, that focuses on the symbolic learning part of the pipeline by using both Logic Tensor Networks and interpretable rule ensembles. After showing that the proposed methodology is able to deliver highly accurate and explainable models, we then discuss potential implementation issues and future directions that can be explored.

Keywords: Explainable AI, Symbolic Machine Learning, Deep Learning, Kandinsky Patterns

1 Introduction

In the recent years, there has been a significant growth in the popularity of Machine Learning (ML) solutions, mainly driven by the increasing success of a specific type of ML, i.e., Deep Learning (DL), in a number of various applications: from game playing [41,70] and natural language processing [81], to self-driving vehicles [58] and computer-assisted medicine [21,32,33,36]. These two latter domains shed light on what has been one of the more severe limitations of the DL methodology, that is its black-box nature and lack of explainability [13], a topic that is becoming of primary importance, also in light of recent regulations like the GDPR [17,29], which stipulates that any significant or legally related decision should be explainable if reached in non-supervised automated processes.

This limitation should be considered along a twofold perspective: from a decision-support perspective, because decision makers are typically required to be accountable for their decisions in critical domains (like juridic and medical settings) and give indications about their interpretations and judgments; and

from a system-oriented perspective, because the lack of explainability makes it difficult to reason about the robustness and actual skills of a ML system (and the socio-technical system relying on its operation), as it is shown by phenomena like adversarial examples [28], misguided usage of context information [61], or general data quality issues [11,12].

In order to address the above limitations, many approaches toward *eXplainable AI* (XAI) have been proposed and discussed [31] and different proposals to evaluate the explainability and causability [37,38] of ML models have been developed [39]. The techniques to achieve explainability can be distinguished in two broad categories: approaches that are based on the development of *instrinsically* interpretable ML models (e.g. decision rules [44,79], decision trees or linear classifiers [77]); or so-called *post-hoc* approaches, whose goal is to *make* an already existing model *understandable*, either through methods that explain the general model behaviour (e.g. using interpretable surrogate models [9] or visualization techniques, such as *saliency maps* [71]), or through local explanation techniques that only attempt to explain how the ML model arrived at its conclusion *for a specific instance* [30,61].

Drawing on recent research [3,4,26] that shows some relevant limitations in post-hoc explainability approaches, we argue that these approaches toward XAI are currently insufficient, also because they do not enable a true understanding of the causal properties of the ML models they are meant to explain [37]. More generally, we posit that the major obstacles toward building truly explainable AI systems reside in two properties of how Deep Learning is *currently used*: first, the end-to-end training process that, while allowing the development of highly accurate models, results in the discovery of *features* which are typically not guaranteed to be understandable by humans [34]; second, their essentially propositional nature, which contrasts with the fact that human knowledge is usually relational [35].

The main goal of this position paper is to put forward a perspective toward tackling the two above-mentioned issues, based on an hybrid subsymbolic-symbolic learning paradigm, a framework reconciling ML and Knowledge Representation & Reasoning (KRR) that has been attracting increasing interest and has recently been advocated as a way-forward for the field of Artificial Intelligence [18,20], also due to advancements in Statistical Relational Learning (SRL) [57] and Neuro-Symbolic (NeSy) [24,52] computation. Under this framework, we promote an integration of Deep Learning and symbolic ML techniques, in order to profit of the advantages of both methodologies. Specifically, going back to the notion of *feature engineering* [82], we propose to use the superior pattern recognition performance of Deep Learning in order to automatize the detection of *high-level, hand-crafted features*, that ideally should also be *verifiable* (e.g. object detection in image classification tasks), which are then to be employed for the training of highly interpretable symbolic models.

In the rest of this paper we discuss this proposal, specifically in Section 2, after providing a background of relevant DL and Explainable AI techniques, we describe the proposed methodology. We then also present a prototypical example,

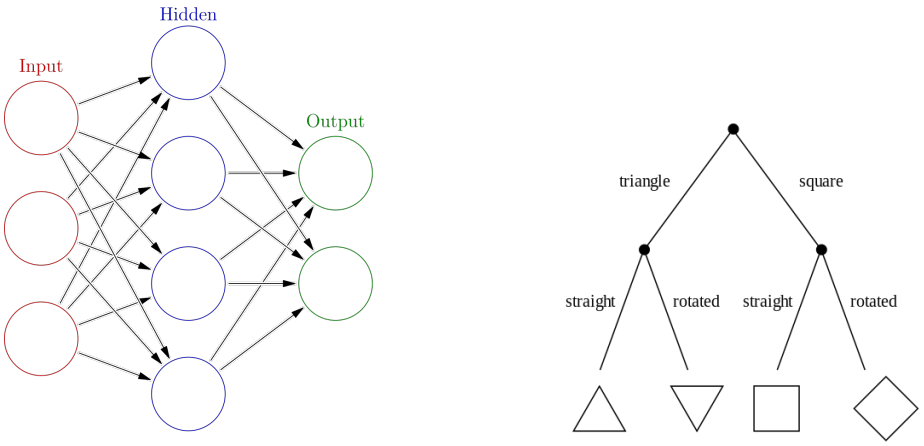
focusing on the symbolic learning part of the proposed framework through the usage of Logic Tensor Networks (LTN) [68] and rule ensembles, in the context of two datasets generated from the recently proposed *Kandinsky Patterns* [39] benchmark for XAI. The results of these experiments are reported in Section 4. Finally, in Section 5 we discuss the obtained results, the implications of the proposed methodologies, its current advantages and limitations and possible directions for further exploration.

2 Methods

2.1 Background

In this section we briefly recall basic notions about Deep Learning architectures and the black box problem, then we present a brief introduction to XAI techniques.

Deep Learning and Neural Networks Deep Learning [27] is a ML paradigm, based on Artificial Neural Networks (ANN), which involves fitting an highly non-linear, differentiable, parametric model defined by the composition of non-linear functions collected in layers (see Figure 1a for an example of such models).



(a) An example of a deep learning model, with a single hidden layer model, visualized as a network.

(b) An example of a Decision Tree, one of the most common transparent ML models.

Fig. 1: Examples of black box and interpretable models.

Despite being first studied as far as the late '60s [40,48] and in its modern form from the '80s [23,65], this paradigm received a boost in popularity starting from

2012 [14,43], due to advancements in hardware technology [56], optimization and regularization methods [42,76], the availability of large amount of data, and its increasing success in real-world applications, especially in regard to image recognition and natural language processing tasks.

The training of Deep Learning models involves the optimization of the parameters of the underlying network (i.e. the weights connecting the units in different layers) with respect to a loss function, typically using local search techniques such as stochastic gradient descent.

A particularly effective Deep Learning architecture for image classification tasks is represented by Convolutional Neural Networks. These architectures are composed by an alternation of convolutional layers (in which regions of the input image are processed by trainable filters which are supposed to detect and isolate low-level features) and pooling layers (which are used for dimensionality reduction).

Despite the huge success of Deep Learning architectures they have recently been criticized for their *black box* nature and lack of explainability: despite achieving high accuracy, the model learned by a DL architecture is often imperscrutable (due to the complex interactions between the high number of non-linear functions) [13] and possibly subject to a variety of attacks [28]: both problems (also in combination) could hinder the actual applicability of these systems in critical application domains, like medicine.

Explainable AI In order to address these limitations, an increasing interest has recently been placed on the development of so-called Explainable AI techniques [55]. We can distinguish between post-hoc explainability techniques and intrinsic explainability (or transparency).

Post-hoc explainability refers to methodologies that aim to make an already existing and trained black box model explainable, or interpretable. These methodologies can be further differentiated between *global* and *local* methodologies. In the former case, the goal is to train an interpretable model that provides a good approximation of the underlying black box model. To this aim, the most common techniques are: Decision Tree induction [78,80], rule extraction methods [10,69] or architecture-specific methodologies such as Layer-wise Relevance Propagation for CNNs [54]. Although a vast number of these techniques have been proposed in the literature, sometimes with high approximation accuracy, their applicability to cases with large numbers of features or with low-level features (e.g. in image-related tasks) have recently been questioned [2].

Local explainability, on the other hand, tries to produce justifications why the black box model made a specific decision for *specific instances*. Also in this case, different methods have been proposed, such as LIME [61], or the related *anchors* [62] *Leave-One-Covariate-Out* [47], which ground on the idea of finding relevant features and associated decision rules for explaining single predictions, or approaches for local interpretability of CNN models such as saliency (or activation) maps [71]. Despite the great interest in these techniques, and some recent efforts to provide a theoretical understanding and to find mutual relationships

among different local explainability approaches [51], recent research has casted some doubt on their real relevance and robustness. For instance, Mittelstadt et al. [53] criticize post-hoc explainability approaches and argue for a broader perspective on explainability; Barocas et al. and Laugel et al. [5,46,45] recently highlighted hidden assumptions required for feature-based local explainability techniques to properly work; Slack et al. [73] showed how LIME and related methods can be subject to adversarial attacks making them hide potential biases in the underlying black model; Sixt et al. [72] recently questioned the explaination capacity of attribution methods for CNNs (such as Layer-wise Relevance Propagation and saliency maps).

In light of these criticisms towards post-hoc explainability methods, calls to avoid the use of black box models augmented with explainability methods in high-stakes domains have recently been made [64], urging for the adoption of transparent (or, intrinsically interpretable) models in these contexts.

In this case, the goal is to directly train and use in the considered domain ML models which are intrinsically transparent and interpretable: such as shallow decision trees [9], Bayesian Networks [8] or rule lists [44,79] (see Figure 1b) or relational approaches such as those based on inductive logic programming [66].

However, the same limitations that apply to global interpretability techniques also apply to transparent models [2], especially in regard to their loss of transparency when the number of relevant feature grows, and in regard to the fact that they are not currently capable to match the performance of DL approaches in tasks such as imaging-related diagnosis.

The approach that we propose, and that we describe in the next section, tries to address both this limitation and the black-box problem of Deep Learning through a combination of these two approaches.

2.2 Proposed Methodology

The approach that we propose, to tackle both the black-box problem of Deep Learning models and the limitations of symbolic transparent models, is based on an integration of the two approaches.

We argue that the main source of the black-box problem in Deep Learning models is not an intrinsic property of the model family or the technique in se, but rather a consequence of how they are typically used.

The former problem is related to the end-to-end approach that is usually employed to train these models. This learning paradigm allows the DL models to learn highly accurate parameter assignments by automatically finding high-level features that arise as complex non-linear combinations of the low-level ones originally present in input representation. The *classifier* part of the DL model is usually implemented as a (jointly trained) simple logistic regression layer on the highest-level features. The learned features, however, are usually inscrutable to a human user.

In contrast, the approach that was traditionally used to achieve high accuracy with standard classifiers (e.g. Support Vector Classifiers or Boosting models) was based on *hand-crafted* informative features [82].

Our approach is based on an integration of these two paradigms: instead of jointly training both a feature detector and a classifier, our proposal consists in directly training a DL model solely as a feature detector for high-level, informative and hand-crafted features.

In particular, we focus on models for feature detection in image recognition tasks. In this case, features represent specific *objects* or *artifacts* that should be detected in the input images: some examples are the presence of fractures (e.g. in bone MRIs), or of nodules or other anomalous objects (e.g. in mastography or similar diagnostic imaging). Compared to the traditional feature engineering (FE) pipeline, this approach has many advantages:

- The human annotators are only required to specify the set of *features* that could be detected in the given domain, and provide annotations for these features in a limited set of training examples that can then be used to train a DL model to automatically detect these features in any future image;
- Compared to the traditional FE pipeline that either employs hand-crafted algorithms or traditional ML models, this approach allows to benefit from the (usually) superior performance of DL models. In fact, specific DL architectures have been developed that could be applied to this task: for example *object detection* or *semantic segmentation* architectures such as Fully Convolutional networks [49], YOLO networks [59] or Faster R-CNN [60], which have been shown to be highly effective in real-world tasks, can be applied to implement this feature-detection task;
- Compared with the traditional end-to-end learning approach for DL models, this approach guarantees that the learned features are both interpretable (as they have previously been identified by human users) and also *testable*. By this term, we mean that their presence and nature could easily be checked in the original images by an external human user: this could be implemented by simply requiring that the DL detection model provides the identified patterns or features, as well their location in the original image (e.g. via a bounding box);
- The feature annotation task to be performed by the DL models is, naturally, a multi-target learning problem: this type of tasks have been shown to act as a regularization method for the DL models (as the shared weights are required to be optimized to several tasks simultaneously) and hence may result in improved generalization and reduce overfitting [63];

The second problem relates to the fact that the classifier used in DL models (usually a simple logic regression layer), while it uses high-level but uninterpretable features, can usually be only understood in terms of very low-level (e.g. pixel-based in imaging tasks) features.

In our approach, on the other hand, the *classifier* component of the learning architecture is separated from the feature detection component, so that more expressive and inherently interpretable learning algorithms can be applied. In this article, we will consider these two approaches, namely *Logic Tensor Networks* and *rule ensembles*.

Logic Tensor Network [68] is a neuro-symbolic computational model integrating neural network and a first-order fuzzy logic called *real logic*. It allows for the combination of the learning capabilities of Deep neural networks with the expressivity and interpretability of logic programming, and it has been shown to be effective in knowledge completion tasks [68], semantic image interpretation [67] and deductive reasoning [7]. The main advantage of this model typology is that, since it is based on a first-order fuzzy logic, it natively allows for the expression of relational concepts and that it is fully compatible with traditional DL models. This latter characteristic could be particularly meaningful in the approach that we propose as it could allow users to recover a sort of end-to-end learning. From a technical point of view, as presented in [68], a Logic Tensor Network is defined by a multi-layer neural network encoding a collection of (prenex, conjunctive Skolemized) clauses expressed in the language of first-order real logic, i.e. logical formulas in the following form:

$$\begin{aligned}&\forall \mathbf{x} = \langle x_1, ..., x_n \rangle. \\ &\quad P_1(f_1^1(\mathbf{x}), ..., f_h^1(\mathbf{x})) \vee ... \vee P_k(f_1^k(\mathbf{x}), ..., f_m^k(\mathbf{x})) \vee \\ &\quad \vee \neg P_{k+1}(f_1^{k+1}(\mathbf{x}), ..., f_r^{k+1}(\mathbf{x})) \vee ... \vee \neg P_l(f_1^l(\mathbf{x}), ..., f_s^l(\mathbf{x})),\end{aligned} \tag{1}$$

where each P_i is a symbol predicate and each f_j^i is a symbol function. The clauses are represented as multi-layer neural network by associating to each symbol predicate P a neural tensor network [75] in the form:

$$\mathbf{N}(P) = \sigma(u_P^T * tanh(\mathbf{x}^T W_P \mathbf{x} + V_P \mathbf{x})), \tag{2}$$

where σ is the sigmoid function, W_P and V_P are tensors. The different networks corresponding to the predicates are then joined (according to a specified clause) by defining $\neg$ to be a fuzzy negation (e.g. $\neg(x) = 1 - x$) and $\vee$ to be a t-conorm (e.g. $\vee(x, y) = max(x, y)$). The parameters of the resulting network are then trained via standard backpropagation by maximizing the satisfiability of the clauses. For a more in-depth introduction to Logic Tensor Network we refer the reader to [68,67].

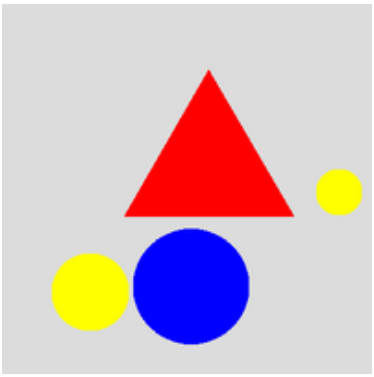
Rule ensembles [22], on the other hand, are based on training an ensemble of rules, where each rule is an expression of the form if $att_1 = val_1 \wedge ...att_n = val_n$ then $class = y$, where att_i represents one of the features and val_i a possible value for that feature, where n (i.e. the number of features involved in the rule) is small. In particular, each rule is trained in order to have high accuracy at detecting a specific class. Rule ensembles allow to obtain interpretable but very robust classifiers, which often achieve a performance that is comparable with other ensemble models (e.g. Random Forest). The main advantage of this model class is that there is a large availability of out-of-the-box computationally efficient techniques and algorithms for training such classifiers based on techniques similar to gradient boosting (e.g. SLIPPER [16] or RuleFit [22]), maximum likelihood estimation [19], Rough Sets [74] or simply by extracting rules from decision trees. Notably, these algorithms are not only computationally efficient but they also have been showed to usually provide good performance even with small sample

sizes. Furthermore, while these models are based on propositional logic (as they are represent as conjunctions over ground feature values, not involving relations among them or quantifiers), simple post-processing steps can be performed to transform the rules into a relational form.

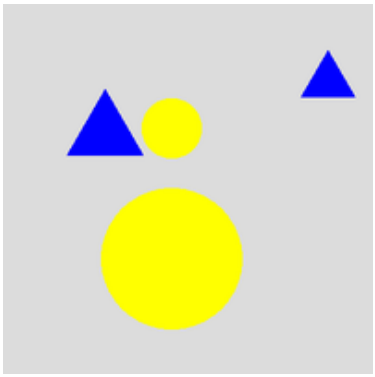
2.3 Kandinsky Patterns and Proof-of-concept Experiments

In order to demonstrate the feasibility and effectiveness of the proposed subsymbolic-symbolic integration approach, with the two specific implementations based on LTN and rule ensembles, we provide a proof-of-concept experimental evaluation of the approach based on two datasets from the Kandinsky Patterns benchmark [39]. Kandinsky Patterns are datasets composed of patterns of geometric shapes described by either first-order logical formulas or mathematical expressions. Each pattern describes a binary classification problem in which the goal is to discriminate between images that belong to the pattern (i.e. positive examples) and images that do not belong to the pattern (i.e. negative examples).

In particular, we considered two benchmark datasets that are shown in Figures 2 and 3. In the first benchmark, denoted as *OneRed* the positive class is composed of images containing at least one red shape. In the second benchmark, denoted as *TwoPairs*, the positive class is composed of images containing exactly four shapes in two pairs: the first pair consists of two objects with same shape and same color, while the second pair consists of two objects with same shape but different colors. We selected these two benchmarks as they do not involve overlapping figures or complex spatial patterns: for this reason, they are relatively easy tasks for a DL-based object recognition model.

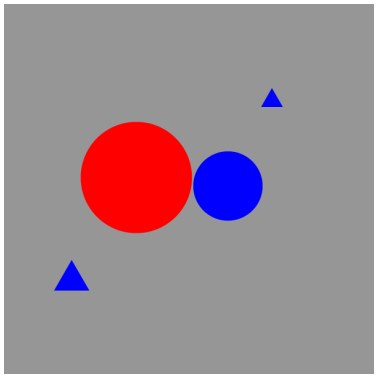


(a) Positive example: its vector-representation is $\langle t, r, c, y, c, y, c, b\rangle$

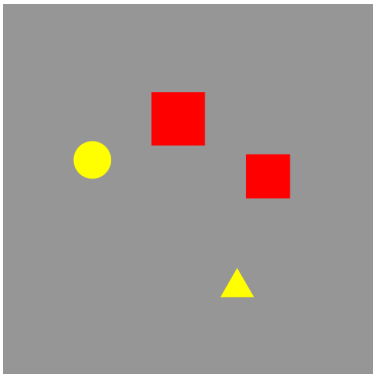


(b) Negative example: its vector-representation is $\langle t, b, t, b, c, y, c, y\rangle$

Fig. 2: The *OneRed* benchmark.



(a) Positive example: its vector-representation is $\langle 0, 0, 0, 1, 0, 2\rangle$



(b) Negative example: its vector-representation is $\langle 0, 2, 0, 0, 0, 0\rangle$

Fig. 3: The *TwoPairs* benchmark.

We considered, for both benchmarks, images with exactly 4 objects in order to adopt a fixed-length vector representation as input of the symbolic models. For the *OneRed* benchmark imags are represented as vectors $\langle sh_1, col_1, ..., sh_4, col_4\rangle$, where

$$sh_i \in \{triangle(t), square(s), circle(c)\} \tag{3}$$

$$col_i \in \{red(r), blue(b), yellow(y)\} \tag{4}$$

On the other hand, for the *TwoPairs* benchmark we adopted an higher-level representation as the task is inherently more complex (as it involves comparison of pairs of objects). This is particularly relevant for the ensemble rule learning: indeed, if we were to employ a representation similar to the one for the *OneRed* benchmark, the learned classifier would be composed of a large number of rules. Thus, each image is represented as a vector $\langle eq_{1,2}, eq_{1,3}, eq_{1,4}, eq_{2,3}, eq_{2,4}, eq_{3,4}\rangle$, where $eq_{i,j} \in \{0, 1, 2\}$ and the semantics of these values is defined as follows:

- $eq_{i,j} = 2$ means that objects i, j have both the same shape and the same color;
- $eq_{i,j} = 1$ means that objects i, j have the same shape but different colors;
- $eq_{i,j} = 0$ means that objects i, j differ in both shape and color.

Thus, for the *OneRed* benchmark the positive examples are those for which

$$\exists i \in \{1, ..., 4\} \text{ s.t. } color_i = red \tag{5}$$

while for the *TwoPairs* benchmark the positive examples are those described by the formula

$$\exists i \neq j \neq k \neq l \text{ s.t. } eq_{i,j} = 2 \wedge eq_{k,l} = 1 \tag{6}$$

We notice that the *OneRed* benchmark consists of 6561 possible different vector encodings (in the adopted representation), while the *TwoPairs* benchmark consists of 29 possible distinct patterns.

3 Implementation

For the *OneRed* and *TwoPairs* benchmark, we generated, respectively, 1000 and 400 different random inputs with balanced classes: in order to avoid overfitting we checked via the generation script that no two images in the dataset were identical. As the main goal of the proof-of-concept experiment was to show the effectiveness of feature annotation combined with symbolic models for obtaining accurate and interpretable models, we directly generated the vector encodings and hence we did not explicitly trained a DL model for the feature annotation starting from images (although, as we previously mentioned, we expect a DL object recognition model to perform effectively in these tasks as they only involve simple, non-overlapping geometric shapes).

In order to train and evaluate the considered models we performed a 75%/25% train-test split of the two datasets.

The Logic Tensor Network models were implemented using the Tensorflow [1] framework with the logictensornetworks[1] API. The models for both benchmarks have been defined by a single predicate (representing the target to be learned) and two axioms establishing the value of predicates on the positive and negative examples of the training set respectively.

On the other hand, as regards rule ensembles, we implemented the model using the skope-rules library[2], which implements a rule ensemble algorithm based on rule extraction from Random Forests estimators.

4 Results

Logic Tensor Network models obtained 90% accuracy for the *OneRed* benchmark and 75% accuracy for the *TwoPairs* benchmarks, on the test set, while the rule ensemble models obtained 100% accuracy for both benchmarks and the learned rules were the correct description of the target class, albeit in propositional form. For the *OneRed* benchmark, the learned rules are:

- $color_1 = red \implies OneRed = 1$;
- $color_2 = red \implies OneRed = 1$;
- $color_3 = red \implies OneRed = 1$;
- $color_4 = red \implies OneRed = 1$.

where by $OneRed = 1$ we mean that the algorithm predicts that the instance is a positive one.

Similarly, for the *TwoPairs* benchmark the learned rules are:

[1] `https://github.com/logictensornetworks/logictensornetworks`
[2] `https://github.com/scikit-learn-contrib/skope-rules`

- $eq_{1,2} = 2 \wedge eq_{3,4} = 1 \implies TwoPairs = 1;$
- $eq_{1,3} = 2 \wedge eq_{2,4} = 1 \implies TwoPairs = 1;$
- $eq_{1,4} = 2 \wedge eq_{2,3} = 1 \implies TwoPairs = 1;$
- $eq_{2,3} = 2 \wedge eq_{1,4} = 1 \implies TwoPairs = 1;$
- $eq_{2,4} = 2 \wedge eq_{3,4} = 1 \implies TwoPairs = 1;$
- $eq_{3,4} = 2 \wedge eq_{1,2} = 1 \implies TwoPairs = 1;$

Since the learned rules correctly represent the target concepts, and there was no overlap between the train and test sets, we can claim that the perfect accuracy obtained by rule ensemble models were not due to overfitting. Furthermore, it is evident that the learned rules are fully interpretable and could be used by a human decision-maker to understand the reasons for classifying a given novel instance.

5 Discussion and Conclusion

The results reported above show that the proposed methodology could be an effective way to integrate DL models, as feature detectors, and symbolic models, in order to obtain interpretable and verifiable models which are, nonetheless, very accurate. Rule ensemble models, in particular, showed to be robust and effective models for the purpose of our proof-of-concept experimentation, as they achieved perfect accuracy while resulting in transparent and understandable models. We notice however, as a limitation, that the considered benchmarks were actually quite simple and this has a clear influence on the achieved model accuracy.

Our goal was twofold: primarily, to show the effectiveness of symbolic models also in perceptual tasks (such as image classification), when they are supplied with relevant high-level features; then, to show the effectiveness of feature engineering to obtain transparent models. For this reason, we did not explicitly trained a model to perform the required feature annotation; this allowed us to ignore the explicit interaction between the output of the DL models and the symbolic models. In the setting of a real experiment, however, this integration step is important and a careful analysis of how this is performed should be necessary, e.g. with reference to what the output format is of the DL feature detectors: would it be preferable a threshold arg-max binarization of the features (as assumed in our proof-of concept), or to directly supply the symbolic models with the un-thresholded output of the final soft-max layer of the network? The former choice would result in more interpretable binary classifiers for the symbolic component of the pipeline; the latter solution, on the otehr hand, may allow for symbolic models that employ noisy information even in cases of mis-detection of the features (whereas, in this case, the arg-max solution would supply the symbolic models with incorrect information). In this latter case, in order to properly manage this uncertain and partial form of information about the features, more expressive symbolic models could be employed, e.g. models based on fuzzy logic or other formalisms for uncertainty management. Similarly, one should consider the level of abstractness of the annotations that the DL model is trained to reproduce: indeed, our example show two different levels of abstractness, ranging

from the simple object-level representation adopted for the *OneRed* benchmark, to the more complex pair-level representation adopted for the *TwoPairs* benchmark. This shows that in the proposed approach there is a measurable trade-off between interpretability, achieved by adopting a more expressive representation for the instances, and annotation easiness, as one should expect lower-level representations to be both more easily elicitable (by the human raters) and also more easy to discriminate (by the DL model).

FInally, another aspect that should be taken in consideration regards the process of feature annotation: in our proof-of-concept, this process was automatically performed via a script during the image generation step. In real-world tasks, we could expect that this process would be performed by expert human annotators in the multi-rater settings (i.e. settings in which multiple human raters annotate a given dataset): as noted in recent research [12], this annotation task could incur in annotation errors that may impact the quality of the ground truths and, hence, the performance of the DL feature detectors trained on them. Also in this case, employing symbolic formalism for uncertainty management may be useful.

Another interesting aspect that should be considered in the multi-rater settings regards how the annotations, and their influence on the related target label, are represented. Indeed, in this work, we employed a simple vector-valued representation of the features all together with the assigned target label, with no explicit relationship between the two (i.e. we employed no direct specification of which annotated feature or feature value were relevant for the target decision). However, one could also conjecture that, in complex annotation tasks, more expressive logic-based formalisms should be employed: e.g. for a given case instance, a human rater could specify that a specific target label is assigned (e.g. the instance is a positive example) because a specific subset of features is present (e.g., in a medical setting, the presence of a specific type of tissue lesion). This setting, which is based on the computational modeling of argumentation [6] and may benefit from recent research in learning from argumentation [15] and integration of DL with argumentation [25], could allow for a more explicit and informative representation of the inter-relationship among the annotated features and the target labels, expressed in the argumentative formalism which have been shown to be effective in complex decision-making settings such as the medical one [50]. For this reason, this could be a relevant further direction to explore.

References

1. Abadi, M., Barham, P., Chen, J., et al.: Tensorflow: A system for large-scale machine learning. In: 12th USENIX OSDI Symposium. pp. 265–283 (2016)
2. Adadi, A., Berrada, M.: Peeking inside the black-box: A survey on explainable artificial intelligence (xai). IEEE Access 6, 52138–52160 (2018)
3. Adebayo, J., Gilmer, J., Goodfellow, I.J., Kim, B.: Local explanation methods for deep neural networks lack sensitivity to parameter values. CoRR abs/1810.03307 (2018)

4. Adebayo, J., Gilmer, J., Muelly, M., Goodfellow, I., Hardt, M., Kim, B.: Sanity checks for saliency maps. In: Advances in Neural Information Processing Systems. pp. 9505–9515 (2018)
5. Barocas, S., Selbst, A.D., Raghavan, M.: The hidden assumptions behind counterfactual explanations and principal reasons. In: Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency. pp. 80–89 (2020)
6. Baroni, P., Gabbay, D.M., Giacomin, M., van der Torre, L.: Handbook of formal argumentation. College Publications (2018)
7. Bianchi, F., Hitzler, P.: On the capabilities of logic tensor networks for deductive reasoning. In: AAAI Spring Symposium: Combining Machine Learning with Knowledge Engineering (2019)
8. Bielza, C., Larrañaga, P.: Discrete bayesian network classifiers: A survey. ACM Comput. Surv. 47 (2014)
9. Blanco-Justicia, A., Domingo-Ferrer, J.: Machine learning explainability through comprehensible decision trees. In: International Cross-Domain Conference for Machine Learning and Knowledge Extraction. pp. 15–26. Springer (2019)
10. Bologna, G., Hayashi, Y.: A comparison study on rule extraction from neural network ensembles, boosted shallow trees, and svms. Applied Computational Intelligence and Soft Computing 2018 (2018)
11. Cabitza, F., Campagner, A., Sconfienza, L.: As if sand were stone. new concepts and metrics to probe the ground on which to build trustable ai. BMC Medical Informatics and Decision Making (2020), submitted
12. Cabitza, F., Ciucci, D., Rasoini, R.: A giant with feet of clay: on the validity of the data that feed machine learning in medicine. In: Organizing for the Digital World, pp. 121–136. Springer (2019)
13. Castelvecchi, D.: Can we open the black box of ai? Nature News 538(7623), 20 (2016)
14. Ciregan, D., Meier, U., Schmidhuber, J.: Multi-column deep neural networks for image classification. In: 2012 IEEE conference on computer vision and pattern recognition. pp. 3642–3649. IEEE (2012)
15. Cocarascu, O., Toni, F.: Argumentation for machine learning: A survey. In: COMMA. pp. 219–230 (2016)
16. Cohen, W.W., Singer, Y.: A simple, fast, and effective rule learner. AAAI/IAAI 99(335-342), 3 (1999)
17. Crockett, K., Goltz, S., Garratt, M.: Gdpr impact on computational intelligence research. In: 2018 International Joint Conference on Neural Networks (IJCNN). pp. 1–7. IEEE (2018)
18. De Raedt, L., Dumančić, S., Manhaeve, R., Marra, G.: From statistical relational to neuro-symbolic artificial intelligence. arXiv preprint arXiv:2003.08316 (2020)
19. Dembczyński, K., Kotłowski, W., Słowiński, R.: Maximum likelihood rule ensembles. In: Proceedings of the 25th international conference on Machine learning. pp. 224–231 (2008)
20. Dubois, D., Prade, H.: Towards a reconciliation between reasoning and learning-a position paper. In: International Conference on Scalable Uncertainty Management. pp. 153–168. Springer (2019)
21. Esteva, A., Kuprel, B., Novoa, R.A., Ko, J., Swetter, S.M., Blau, H.M., Thrun, S.: Dermatologist-level classification of skin cancer with deep neural networks. Nature 542(7639), 115 (2017)
22. Friedman, J.H., Popescu, B.E., et al.: Predictive learning via rule ensembles. The Annals of Applied Statistics 2(3), 916–954 (2008)

23. Fukushima, K.: Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. Biological cybernetics 36(4), 193–202 (1980)
24. Garcez, A.d., Gori, M., Lamb, L.C., Serafini, L., Spranger, M., Tran, S.N.: Neural-symbolic computing: An effective methodology for principled integration of machine learning and reasoning. arXiv preprint arXiv:1905.06088 (2019)
25. Garcez, A.S.d., Gabbay, D.M., Lamb, L.C.: A neural cognitive model of argumentation with application to legal inference and decision making. Journal of Applied Logic 12(2), 109–127 (2014)
26. Gilpin, L.H., Bau, D., Yuan, B.Z., et al.: Explaining explanations: An overview of interpretability of machine learning. In: 2018 IEEE 5th DSAA International Conference. pp. 80–89. IEEE (2018)
27. Goodfellow, I., Bengio, Y., Courville, A.: Deep learning. MIT press (2016)
28. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572 (2014)
29. Goodman, B., Flaxman, S.: European union regulations on algorithmic decision-making and a "right to explanation". AI magazine 38(3), 50–57 (2017)
30. Guidotti, R., Monreale, A., Ruggieri, S., et al.: Local rule-based explanations of black box decision systems. arXiv preprint arXiv:1805.10820 (2018)
31. Guidotti, R., Monreale, A., Ruggieri, S., et al.: A survey of methods for explaining black box models. ACM computing surveys (CSUR) 51(5), 1–42 (2018)
32. Gulshan, V., Peng, L., Coram, M., et al.: Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs. Jama 316(22), 2402–2410 (2016)
33. Haenssle, H., Fink, C., Schneiderbauer, R., et al.: Man against machine: diagnostic performance of a deep learning convolutional neural network for dermoscopic melanoma recognition in comparison to 58 dermatologists. Annals of Oncology 29(8), 1836–1842 (2018)
34. Hagras, H.: Toward human-understandable, explainable ai. Computer 51(9) (2018)
35. Halford, G.S., Wilson, W.H., Phillips, S.: Relational knowledge: the foundation of higher cognition. Trends in cognitive sciences 14(11), 497–505 (2010)
36. Han, S.S., Park, G.H., Lim, W., et al.: Deep neural networks show an equivalent and often superior performance to dermatologists in onychomycosis diagnosis: Automatic construction of onychomycosis datasets by region-based convolutional deep neural network. PloS one 13(1), e0191493 (2018)
37. Holzinger, A., Langs, G., Denk, H., Zatloukal, K., Mueller, H.: Causability and explainability of ai in medicine. Data Mining and Knowledge Discovery 10 (2019)
38. Holzinger, A., Carrington, A., Mueller, H.: Measuring the quality of explanations:the system causability scale (scs) comparing human and machine explanations. KI - Kunstliche Intelligenz 34(2) (2020)
39. Holzinger, A., Kickmeier-Rust, M., Müller, H.: Kandinsky patterns as iq-test for machine learning. In: International Cross-Domain Conference for Machine Learning and Knowledge Extraction. pp. 1–14. Springer (2019)
40. Ivakhnenko, A.G., Lapa, V.G.: Cybernetics and forecasting techniques (1967)
41. Justesen, N., Bontrager, P., Togelius, J., Risi, S.: Deep learning for video game playing. IEEE Transactions on Games (2019)
42. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
43. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. pp. 1097–1105 (2012)

44. Lakkaraju, H., Bach, S.H., Leskovec, J.: Interpretable decision sets: A joint framework for description and prediction. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining. pp. 1675–1684 (2016)
45. Laugel, T., Lesot, M.J., Marsala, C., Detyniecki, M.: Issues with post-hoc counterfactual explanations: a discussion. arXiv preprint arXiv:1906.04774 (2019)
46. Laugel, T., Lesot, M.J., Marsala, C., Renard, X., Detyniecki, M.: The dangers of post-hoc interpretability: Unjustified counterfactual explanations. arXiv preprint arXiv:1907.09294 (2019)
47. Lei, J., G'Sell, M., Rinaldo, A., Tibshirani, R.J., Wasserman, L.: Distribution-free predictive inference for regression. Journal of the American Statistical Association 113(523), 1094–1111 (2018)
48. Linnainmaa, S.: The representation of the cumulative rounding error of an algorithm as a taylor expansion of the local rounding errors. Master's Thesis (in Finnish), Univ. Helsinki pp. 6–7 (1970)
49. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3431–3440 (2015)
50. Longo, L., Hederman, L.: Argumentation theory for decision support in healthcare: A comparison with machine learning. In: International Conference on Brain and Health Informatics. pp. 168–180. Springer (2013)
51. Lundberg, S.M., Lee, S.I.: A unified approach to interpreting model predictions. In: Advances in neural information processing systems. pp. 4765–4774 (2017)
52. Mao, J., Gan, C., Kohli, P., Tenenbaum, J.B., Wu, J.: The neuro-symbolic concept learner: Interpreting scenes, words, and sentences from natural supervision. arXiv preprint arXiv:1904.12584 (2019)
53. Mittelstadt, B., Russell, C., Wachter, S.: Explaining explanations in ai. In: Proceedings of the conference on fairness, accountability, and transparency (2019)
54. Montavon, G., Binder, A., Lapuschkin, S., et al.: Layer-wise relevance propagation: an overview. In: Explainable AI: Interpreting, Explaining and Visualizing Deep Learning, pp. 193–209. Springer (2019)
55. Murdoch, W.J., Singh, C., Kumbier, K., Abbasi-Asl, R., Yu, B.: Interpretable machine learning: definitions, methods, and applications. arXiv preprint arXiv:1901.04592 (2019)
56. Oh, K.S., Jung, K.: Gpu implementation of neural networks. Pattern Recognition 37(6), 1311–1314 (2004)
57. Raedt, L.D., Kersting, K., Natarajan, S., Poole, D.: Statistical relational artificial intelligence: Logic, probability, and computation. Synthesis Lectures on Artificial Intelligence and Machine Learning 10(2), 1–189 (2016)
58. Rao, Q., Frtunikj, J.: Deep learning for self-driving cars: chances and challenges. In: Proceedings of the 1st International Workshop on Software Engineering for AI in Autonomous Systems. pp. 35–38 (2018)
59. Redmon, J., Farhadi, A.: Yolov3: An incremental improvement. arXiv (2018)
60. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: Advances in neural information processing systems. pp. 91–99 (2015)
61. Ribeiro, M.T., Singh, S., Guestrin, C.: "why should I trust you?": Explaining the predictions of any classifier. CoRR abs/1602.04938 (2016)
62. Ribeiro, M.T., Singh, S., Guestrin, C.: Anchors: High-precision model-agnostic explanations. In: Thirty-Second AAAI Conference on Artificial Intelligence (2018)

63. Ruder, S.: An overview of multi-task learning in deep neural networks. arXiv preprint arXiv:1706.05098 (2017)
64. Rudin, C.: Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. Nature Machine Intelligence 1(5), 206–215 (2019)
65. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning representations by back-propagating errors. nature 323(6088), 533–536 (1986)
66. Schmid, U.: Inductive programming as approach to comprehensible machine learning. In: Proceedings of DKB-2018 and KIK-2018 (2018)
67. Serafini, L., Donadello, I., Garcez, A.d.: Learning and reasoning in logic tensor networks: theory and application to semantic image interpretation. In: Proceedings of the Symposium on Applied Computing. pp. 125–130 (2017)
68. Serafini, L., Garcez, A.S.d.: Learning and reasoning with logic tensor networks. In: Conference of the Italian Association for Artificial Intelligence. pp. 334–348. Springer (2016)
69. Setiono, R., Baesens, B., Mues, C.: Recursive neural network rule extraction for data with mixed attributes. IEEE Transactions on Neural Networks 19(2), 299–307 (2008)
70. Silver, D., Schrittwieser, J., Simonyan, K., et al.: Mastering the game of go without human knowledge. Nature 550(7676), 354–359 (2017)
71. Simonyan, K., Vedaldi, A., Zisserman, A.: Deep inside convolutional networks: Visualising image classification models and saliency maps. arXiv preprint arXiv:1312.6034 (2013)
72. Sixt, L., Granz, M., Landgraf, T.: When explanations lie: Why many modified bp attributions fail (2019)
73. Slack, D., Hilgard, S., Jia, E., Singh, S., Lakkaraju, H.: Fooling lime and shap: Adversarial attacks on post hoc explanation methods. In: Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society. pp. 180–186 (2020)
74. Ślęzak, D., Widz, S.: Rough-set-inspired feature subset selection, classifier construction, and rule aggregation. In: International Conference on Rough Sets and Knowledge Technology. pp. 81–88. Springer (2011)
75. Socher, R., Chen, D., Manning, C.D., Ng, A.: Reasoning with neural tensor networks for knowledge base completion. In: Advances in neural information processing systems. pp. 926–934 (2013)
76. Srivastava, N., Hinton, G., Krizhevsky, A., et al.: Dropout: a simple way to prevent neural networks from overfitting. Journal of Machine Learning Research 15(1), 1929–1958 (2014)
77. Ustun, B., Rudin, C.: Methods and models for interpretable linear classification. arXiv preprint arXiv:1405.4047 (2014)
78. Van Assche, A., Blockeel, H.: Seeing the forest through the trees: Learning a comprehensible model from an ensemble. In: European Conference on Machine Learning. pp. 418–429. Springer (2007)
79. Wang, T., Rudin, C., Doshi-Velez, F., Liu, Y., Klampfl, E., MacNeille, P.: A bayesian framework for learning rule sets for interpretable classification. The Journal of Machine Learning Research 18(1), 2357–2393 (2017)
80. Yang, C., Rangarajan, A., Ranka, S.: Global model interpretation via recursive partitioning. In: 2018 IEEE 4th DSS Conference. pp. 1563–1570. IEEE (2018)
81. Young, T., Hazarika, D., Poria, S., et al.: Recent trends in deep learning based natural language processing. IEEE Comp Intel Magazine 13(3), 55–75 (2018)
82. Zheng, A., Casari, A.: Feature engineering for machine learning: principles and techniques for data scientists. " O'Reilly Media, Inc." (2018)