ASTERYX : A model-Agnostic SaT-basEd appRoach for sYmbolic and score-based eXplanations

Ryma Boumazouza, Fahima Cheikh-Alili, Bertrand Mazure, Karim Tabia

ASTERYX: A model-Agnostic SaT-basEd appRoach for sYmbolic and score-based eXplanations (Preprint version)

Ryma Boumazouza
boumazouza@cril.fr
Univ. Artois, CNRS, CRIL
Lens, France

Fahima Cheikh-Alili
cheikh@cril.univ-artois.fr
Univ. Artois, CNRS, CRIL
Lens, France

Bertrand Mazure
bertrand.mazure@univ-artois.fr
Univ. Artois, CNRS, CRIL
Lens, France

Karim Tabia
karim.tabia@univ-artois.fr
Univ. Artois, CNRS, CRIL
Lens, France

ABSTRACT

The ever increasing complexity of machine learning techniques used more and more in practice, gives rise to the need to explain the outcomes of these models, often used as black-boxes. Explainable AI approaches are either numerical feature-based aiming to quantify the contribution of each feature in a prediction or symbolic providing certain forms of symbolic explanations such as *counterfactuals*. This paper proposes a generic agnostic approach named ASTERYX allowing to generate both symbolic explanations and score-based ones. Our approach is declarative and it is based on the encoding of the model to be explained in an equivalent symbolic representation. This latter serves to generate in particular two types of symbolic explanations which are *sufficient reasons* and *counterfactuals*. We then associate scores reflecting the relevance of the explanations and the features w.r.t to some properties. Our experimental results show the feasibility of the proposed approach and its effectiveness in providing symbolic and score-based explanations.

CCS CONCEPTS

• **Computing methodologies** → **Artificial intelligence**; *Machine learning*.

KEYWORDS

XAI; Symbolic explanations; Score-based explanation; Model-Agnostic; Satisfiability testing

ACM Reference Format:
Ryma Boumazouza, Fahima Cheikh-Alili, Bertrand Mazure, and Karim Tabia. 2021. ASTERYX: A model-Agnostic SaT-basEd appRoach for sYmbolic and score-based eXplanations (Preprint version). In *Proceedings of the 30th ACM Int'l Conf. on Information and Knowledge Management (CIKM '21), November 1–5, 2021, Virtual Event, Australia.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3459637.3482321

CIKM '21, November 1–5, 2021, Virtual Event, Australia.

ACM ISBN 978-1-4503-8446-9/21/11…$15.00
https://doi.org/10.1145/3459637.3482321

1 INTRODUCTION

In the last decades, the growth of data and widespread usage of Machine Learning (ML) in multiple sensitive fields (e.g. healthcare, criminal justice) and industries emphasized the need for explainability methods. These latter can be grouped into pre-model (ante-hoc), in-model, and post-model (post-hoc). We mainly focus on post-hoc methods where we distinguish two types of explanations: (1) symbolic explanations (e.g. [27],[17]) that are based on symbolic representations used for explanation, verification and diagnosis purposes ([22],[25],[17]), and (2) numerical feature-based methods that provide insights into how much each feature contributed to an outcome (e.g. SHAP[20], LIME[23]). Intuitively, these two categories of approaches try to answer two different types of questions: Symbolic explanations tell why a model predicted a given label for an instance (eg. sufficient reasons) or what would have to be modified in an input instance to have a different outcome (counterfactuals). Numerical approaches, on the other hand, attempt to answer the question to what extent does a feature influence the prediction.

The existing symbolic explainability methods are model-specific (can only be applied to specific models for which they are intended) and cannot be applied agnostically to any model, which is their main limitation. In the other hand, feature-based methods such as Local Interpretable Model-Agnostic Explanations (LIME)[23] and SHapley Additive exPlanations (SHAP)[20] provide the features' importance values for a particular prediction. These values provide an overall information on the contribution of features individually but do not really allow answering certain questions such as: "*What are the feature values which are sufficient in order to trigger the prediction whatever the values of the other variables?* " or "*Which values are sufficient to change in the instance x to have a different prediction?*". This type of questions is fundamental for the understanding, and, above all, for the explanations to be usable. For example, if a user's application is refused, the user will naturally ask the question: "*What must be changed in my application to be accepted?* ". We cannot answer this question in a straightforward manner with the features-based explanations. Thus, the major objective of our

contribution is to provide both symbolic explanations and score-based ones for a better understanding and usability of explanations. It is declarative and does not require the implementation of specific algorithms since its based on well-known Boolean satisfiability concepts, allowing to exploit the strengths of modern SAT solvers. We model our explanation enumeration problem and use modern SAT technologies to enumerate the explanations. The approach provides two complementary types of symbolic explanations for the prediction of a data instance x: ***Sufficient Reasons*** (SR_x for short) and ***Counterfactuals*** (CF_x for short). In addition, it provides score-based explanations allowing to assess the influence of each feature on the outcome. The main contributions of our paper are :

(1) A **declarative** and **model-agnostic** approach allowing to provide SR_x and CF_x explanations based on SAT technologies ;
(2) A set of fine-grained properties allowing to analyze and select explanations and a set of scores allowing to assess the relevance of explanations and features w.r.t the suggested properties ;
(3) An experimental evaluation providing an evidence of the feasibility and efficiency of the proposed approach ;

2 PRELIMINARIES AND NOTATIONS

Let us first formally recall some definitions used in the remainder of this paper. For the sake of simplicity, the presentation is limited to binary classifiers with binary features. We explain negative predictions where the outcome is 0 within the paper but the approach applies similarly[1] to explain positive predictions.

Definition 2.1. **(Binary classifier)** A Binary classifier is defined by two sets of binary variables: A feature space X= $\{X_1,..,X_n\}$ where $|X|$=n, and a binary class variable denoted Y.

A decision function describes the classifier's behavior independently from the way it is implemented. We define it as a function $f : X \rightarrow Y$ mapping each instantiation x of X to y=$f(x)$. A data instance x is the feature vector associated with an instance of interest whose prediction from the ML model is to be explained. We use interchangeably in this paper f to refer to the classifier and its decision function.

Definition 2.2. **(SAT : The Boolean Satisfiability problem)** Usually called SAT, the Boolean satisfiability problem is the decision problem, which, given a propositional logic formula, determines whether there is an assignment of propositional variables that makes the formula true.

The logic formulae are built from propositional variables and Boolean connectors "AND" ($\wedge$), "OR" ($\vee$), "NOT" ($\neg$). A formula is satisfiable if there is an assignment of all variables that makes it true. It is said inconsistent or unsatisfiable otherwise. For example, the formula $(x_1 \wedge x_2) \vee \neg x_1$ where x_1 and x_2 are Boolean variables, is satisfiable since if x_1 takes the value false, the formula evaluates to true. A complete assignment of variables making a formula true is called a model while a complete assignment making it false is called a counter-model.

[1] will be discussed in the "Concluding remarks and discussions" Section.

Definition 2.3. **(CNF (Clausal Normal Form))** A CNF is a set of clauses seen as a conjunction. A clause is a formula composed of a disjunction of literals. A literal is either a Boolean variable p or its negation $\neg p$. A quantifier-free formula is built from atomic formulae using conjunction $\wedge$, disjunction $\vee$, and negation $\neg$. An interpretation μ assigns values from $\{0, 1\}$ to every Boolean variable. Let Σ be a CNF formula, μ satisfies Σ iff μ satisfies all clauses of Σ.

Over the last decade, many achievements have been made to modern SAT solvers[2] that can handle now problems with several million clauses and variables, allowing them to be efficiently used in many applications. Note that we rely on SAT-solving to explain a black-box model where we encode the problems of generating our symbolic explanations as two common problems related to satisfiability testing which are enumerating *minimal reasons* why a formula is inconsistent and *minimal changes to a formula* to restore the consistency. Indeed, in the case of an unsatisfiable CNF, we can analyze the inconsistency by enumerating sets of clauses causing the inconsistency (called Minimal Unsatisfiable Subsets and noted MUS for short), and other sets of clauses allowing to restore its consistency (called Minimal Correction Subsets, MCS for short). The enumeration of MUS/MCS are well-known problems dealt with in many areas such as knowledge base reparation. Several approaches and tools have been proposed in the SAT community for their generation (e.g. [11, 19]).

3 ASTERYX: A GLOBAL OVERVIEW

Our approach is based on associating a symbolic representation that is (almost) equivalent to the decision function of the model to explain. An overview of our approach is depicted on Figure 1.
Given a classifier f, our approach proceeds as follows:

- **Step 1 (Encoding into CNF the classifier)**: This comes down to associating an *equivalent* symbolic representation Σ_f to f. Σ_f will serve to generate symbolic explanations in the next step. The encoding is done either using model encoding algorithms if available and if the encoding is tractable, or using a surrogate approach as described in Section 4.
- **Step 2 (SAT-based modeling of the explanation enumeration problem)**: Once we have the CNF representation Σ_f and the input instance x whose prediction by f is to be explained, we model the explanation generation task as a partial maximum satisfiability problem, also known as Partial Max-SAT [6]. This step, presented in Section 5, aims to provide two types of symbolic explanations: SR_x and CF_x. They respectively correspond to Minimal Unsatisfiable Subsets (MUS) and Minimal Correction Subsets (MCS) in the SAT terminology.
- **Step 3 (Explanation and feature relevance scoring)**: This step aims to assess the relevance of explanations by associating scores evaluating those explanations with regard to a set of properties presented in Section 6. Moreover, this step allows to assess the relevance of features using scoring functions and to evaluate their individual contributions to the outcome.

[2] A SAT solver is a program for deciding the satisfiability of Boolean formulae encoded in conjunctive normal form.

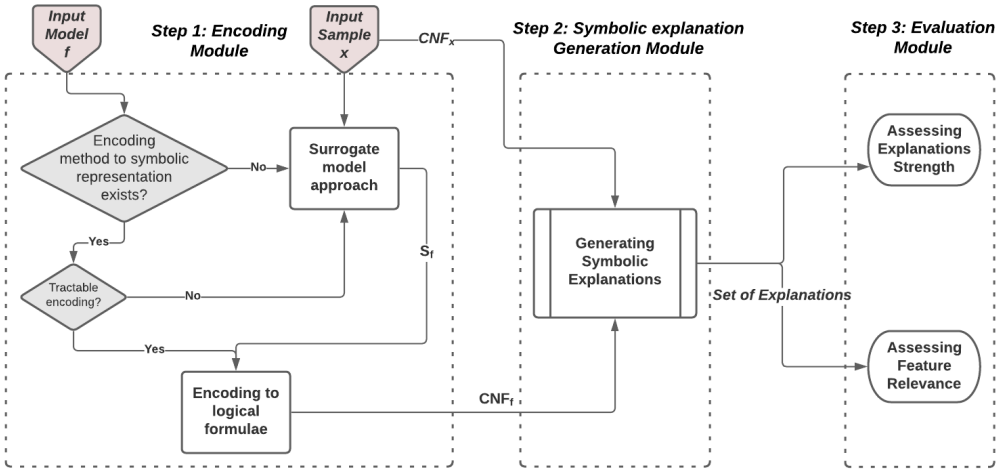


Figure 1: A global overview of the proposed approach

The following sections provide insights for each step of our approach.

4 ENCODING THE CLASSIFIER INTO CNF

This corresponds to **Step 1** in our approach and it aims to encode the input ML model f into CNF in order to use SAT-solving to enumerate our symbolic explanations. Two cases are considered: Either an encoding of classifier f into an equivalent symbolic representation exists (non agnostic case), in which case we can use it, or we consider the classifier f as a black-box and we use a surrogate model approach to approximate it in the vicinity of the instance to explain x (agnostic case). A direct encoding of the classifier f into CNF is possible for some machine learning models such as Binarized Neural Networks (BNNs) [21] and Naive and Latent-Tree Bayesian networks [28]. We mainly focus in this paper on the agnostic option used when no direct CNF encoding exists for f or if the encoding is intractable.

4.1 Surrogate model encoding into CNF

We propose an approach using a surrogate model f_S which is i) as faithful as possible to the initial model f (ensures same predictions) and ii) allows to obtain a tractable CNF encoding. More precisely, we use the surrogate model f_S to approximate the classifier f in the neighborhood of the instance to be explained. Note that one can approximate the classifier f on the whole data set if this latter is available. A machine learning model that can guarantee a good trade-offs between faithfulness and giving a tractable CNF encoding is the one of random forests [12]. As we will see in our experimental study, random forests allow to obtain a good level of faithfulness (in general around 95%) while giving compact CNF encodings in terms of the number of clauses and variables. Given a data instance x whose prediction by the original model f is to be explained and a data set, we construct the neighborhood of x, noted $V(x, r)$, by sampling data instances within a radius r of x. In case the data set is not available, we can draw new perturbed samples around x. Once the vicinity of x sampled, we train a random forest on the data set composed of $(x_i, f(x_i))$ for i=1..p where x_i is a sampled data instance, p is the number of sampled instances. Each x_i is labeled with the prediction $f(x_i)$ since the aim is to ensure that the surrogate model f_S is locally (in x's neighborhood) faithful to f.

Example 4.1. As a running example to illustrate the different steps, we trained a Neural Network model f on the United Stated Congressional Voting Records Data Set[3]. In this example, the label *Republican* corresponds to a positive prediction, noted 1 while the label *Democrat* corresponds to a negative prediction, noted 0. The trained Neural Network model f achieves 95.74% accuracy. An input x consists of the following features :

X_1	handicapped-infants	X_9	mx-missile
X_2	water-project-cost-sharing	X_{10}	immigration
X_3	adoption-of-the-budget-resolution	X_{11}	synfuels-corporation-cutback
X_4	physician-fee-freeze	X_{12}	education-spending
X_5	el-salvador-aid	X_{13}	superfund-right-to-sue
X_6	religious-groups-in-schools	X_{14}	crime
X_7	anti-satellite-test-ban	X_{15}	duty-free-exports
X_8	aid-to-nicaraguan-contras	X_{16}	export-administration-act-south-africa

Assume an input instance x=(1,1,1,0,0,0,1,1,1,0,0,0,0,1,0,1) whose prediction is to be explained. As a surrogate model, we trained a random forest classifier RF_f composed of 3 decision trees (decision tree 1 to 3 from left to right in Fig. 2) on the vicinity of the input sample x. In this example, RF_f achieved an accuracy of 91.66% (RF_f is said locally faithful to f as it has a high accuracy in the vicinity of the instance x to explain).

The CNF encoding of a classifier f (or its surrogate f_S) should guarantee the equivalence of the two representations stated as follows :

Definition 4.2. **(Equivalence of a classifier and its CNF encoding)** A binary classifier f (resp. f_S) is said to be equivalently encoded as a CNF Σ_f (resp. Σ_{f_S}) if the following condition is fulfilled: $f(x)$=1 (resp. $f_S(x)$=1) iff x is a model of Σ_f (resp. Σ_{f_S}).

Namely, data instances x predicted positively ($f(x)$=1) by the classifier are models of the CNF encoding the classifier. Similarly,

[3] Available at https://archive.ics.uci.edu/ml/datasets/congressional+voting+records.

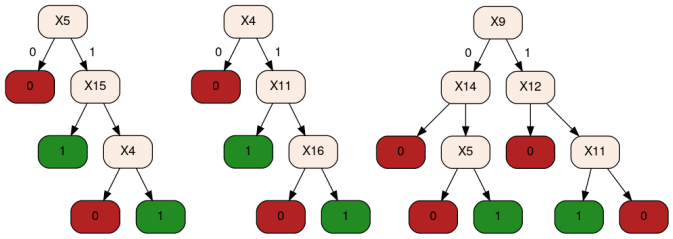


Figure 2: A random forest trained on the neighborhood of x

data instances x predicted negatively ($f(x)$=0) are counter-models of the CNF encoding the classifier.

4.2 CNF encoding of random forests

When we adopt the surrogate approach and use a random forest f_S to agnostically approximate a classifier f, encoding the random forest in CNF amounts to encoding the decision trees individually and then encoding the combination rule (majority voting rule).

- **Encode in CNF every decision tree:** The internal nodes of a decision tree DT_i represent a binary test on one of the features[4]. The leaves of a decision tree, each is annotated with the predicted class (namely, 0 or 1). A decision tree in our case represents a Boolean function.

As shown on Example 4.3, the Boolean function encoded by a decision tree can be captured in CNF as the conjunction of the negation of paths leading from the root node to leaves labelled 0.

- **Encode in CNF the combination rule:** Let y_i be a Boolean variable capturing the truth value of the CNF associated to a decision tree DT_i. Hence, the majority rule used in random forests to combine the predictions of m decision trees can be seen as a cardinality constraint[5] [29] that can be stated as follows:

$$y \Leftrightarrow \sum_{i=1..m} y_i \geq t, \tag{1}$$

where t is a threshold (usually $t=\frac{m}{2}$). Cardinality constraints have many CNF encodings (e.g. [1, 3, 29]). To form the CNF corresponding to the entire random forest, it suffices to conjuct the m CNFs associated to the equivalences between y_i and the CNF of the decisions trees, and, the CNF of the combination rule.

Example 4.3 (Example 4.1 continued). Let us continue with the random forest classifier of Example 4.1. The following formulae illustrate the encoding steps applied to RF_f:

[4]Remember that all the features in our case are binary.
[5]In our case this constraint means that at least t decision trees predicted the label 1.

DT_1 $\quad y_1 \Leftrightarrow (X_5) \wedge (\neg X_5 \vee \neg X_{15} \vee X_4)$

DT_2 $\quad y_2 \Leftrightarrow (X_4) \wedge (\neg X_4 \vee \neg X_{11} \vee X_{16})$

DT_3 $\quad y_3 \Leftrightarrow (X_9 \vee X_{14}) \wedge (X_9 \vee \neg X_{14} \vee X_5) \wedge (\neg X_9 \vee X_{12}) \wedge (\neg X_9 \vee \neg X_{12} \vee \neg X_{11})$

Majority vote $\quad y \Leftrightarrow (y_1 \wedge y_2) \vee (y_1 \wedge y_3) \vee (y_2 \wedge y_3) \vee (y_1 \wedge y_2 \wedge y_3)$

In this example, each decision tree (DT_i, i=1..3) represents a Boolean function whose truth value is captured by Boolean variable y_i. The random forest RF_f Boolean function is captured by the variable y. Note that the encoding of RF_f is provided in this example in propositional logic in order to avoid heavy notations. Direct encoding to CNF could easily be obtained using for example Tseitin Transformation [30].

5 GENERATING SUFFICIENT REASONS AND COUNTERFACTUAL EXPLANATIONS

In this section, we present SR_x and CF_x as well as the SAT-based setting we use to generate such explanations where the input is the CNF encoding of a classifier Σ_f and an input data instance Σ_x whose prediction is to be explained.

5.1 STEP 2: A SAT-based setting for the enumeration of symbolic explanations

Recall that we are interested in two complementary types of symbolic explanations: the *sufficient reasons* (SR_x) which lead to a given prediction and the *counterfactuals* (CF_x) allowing to know minimal changes to apply on the data instance x to obtain a different outcome. Our approach to enumerate these two types of explanations is based on two very common concepts in SAT which are MUS and MCS that we will define formally in the following. To restrict the explanations only to clauses that concern the input data x and do not include clauses that concern the encoding of the classifier, we use a variant of the SAT problem called Partial-Max SAT [6] which can be efficiently solved by the existing tools implementing the enumeration of MUSes and MCSes such as the tool in [10].

A Partial Max-SAT problem is composed of two disjoint sets of clauses where Σ_H denotes the hard clauses (those that could not be relaxed) and Σ_S denotes the soft ones (those that could be relaxed). In our modeling, the set of hard clauses corresponds to Σ_f and the soft clauses to Σ_x representing the CNF encoding of the data instance x whose prediction $f(x)$ is to be explained. Let Σ_x be the *soft clauses*, defined as follows :

- Each clause $\alpha \in \Sigma_x$ is composed of exactly one literal ($\forall \alpha \in \Sigma_x, |\alpha| = 1$).
- Each literal representing a Boolean variable of Σ_x corresponds to a Boolean variable $\{X_i \in X\}$.

Recall that since the classifier f is equivalently encoded to Σ_f, then a negative prediction $f(x)$=0 corresponds to an unsatisfiable CNF $\Sigma_f \cup \Sigma_x$. Now, given an unsatisfiable CNF $\Sigma_f \cup \Sigma_x$, it is possible

to identify the subsets of Σ_x responsible for the unsatisfiability (corresponding to reasons of the prediction $f(x)$=0), or the ones allowing to restore the consistency of $\Sigma_f \cup \Sigma_x$ (corresponding to *counterfactuals* allowing to flip the prediction and get $f(x)$=1).

5.2 Sufficient Reason Explanations (SR_x)

We are interested here in identifying minimal reasons why the prediction is $f(x)$=0. This is done by identifying subsets of clauses causing the inconsistency of the CNF $\Sigma_f \cup \Sigma_x$ (recall that the prediction $f(x)$ is captured by the truth value of $\Sigma_f \cup \Sigma_x$). Such subsets of clauses encoding the input x are *sufficient reasons* for the prediction being negative. We formally define the $\boldsymbol{SR_x}$ explanations as follow:

Definition 5.1. **($\boldsymbol{SR_x}$ explanations)** Let x be a data instance and $f(x)$=0 its prediction by the classifier f. A sufficient reason explanation $\tilde{x}$ of x is such that:

i. $\tilde{x} \subseteq x$ ($\tilde{x}$ is a part of x)
ii. $\forall \acute{x}, \tilde{x} \subset \acute{x} : f(\acute{x})=f(x)$ ($\tilde{x}$ suffices to trigger the prediction)
iii. There is no partial instance $\hat{x} \subset \tilde{x}$ satisfying i and ii (minimality)

Intuitively, a *sufficient reason* $\tilde{x}$ is defined as the part of the data instance x such that $\tilde{x}$ is minimal and causes the prediction $f(x)$=0. We now define Minimal Unsatisfiable Subsets :

Definition 5.2. (**MUS**) A Minimal Unsatisfiable Subset (MUS) is a minimal subset Γ of clauses of a CNF Σ such that $\forall\ \alpha \in \Gamma$, $\Gamma \backslash \{\alpha\}$ is satisfiable.

Clearly, a MUS for $\Sigma_f \cup \Sigma_x$ comes down to a subset of soft clauses, namely a part of x that is causing the inconsistency, hence the prediction $f(x)$=0.

Proposition 5.3. *Let f be a classifier, let Σ_f be its CNF representation. Let also x be a data instance predicted negatively ($f(x)$=0) and let $\Sigma_f \cup \Sigma_x$ be the corresponding Partial Max-SAT encoding. Let $SR(x, f)$ be the set of sufficient reasons of x wrt. f. Let $MUS(\Sigma_{f,x})$ be the set of MUSes of $\Sigma_f \cup \Sigma_x$. Then:*

$$\forall \tilde{x} \subseteq x, \tilde{x} \in SR(x, f) \iff \tilde{x} \in MUS(\Sigma_{f,x}) \quad (2)$$

Proposition 5.3 states that each MUS of the CNF $\Sigma_f \cup \Sigma_x$ is a SR_x for the prediction $f(x)$=0 and vice versa. The proof is straightforward. It suffices to remember that the decision function of f is equivalently encoded by Σ_f and that the definition of a MUS on $\Sigma_f \cup \Sigma_x$ corresponds exactly to the definition of an SR_x for $f(x)$.

Example 5.4 (Example 4.3 continued). Given the CNF $\Sigma_f \cup \Sigma_x$ associated to RF_f from Example 4.3 and the input x=(1, 1, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 0, 1), we enumerate the SR_x for $f(x)$=0 (x is predicted as *Democrat*). There are three SR_x:

- SR_x1="X_4=0 AND X_5=0" (meaning that if the features *physician-fee-freeze* (X_4) and *el-salvador-aid* (X_5) are set to 0, then the prediction is 0) ;
- SR_x2="X_{12}=0 AND X_5=0" ;
- SR_x3="X_4=0 AND X_{12}=0 AND X_9=1" ;

It is easy to check for instance that if X_4=0 and X_5=0 then DT_1 and DT_2 of Fig. 2 predict 0 leading the random forest to predict 0.

5.3 Counterfactual Explanations (CF_x)

For many applications, knowing the reasons for a prediction is not enough, and one may need to know what changes in the input need to be made to get an alternative outcome. Let us formally define the concept of counterfactual explanation.

Definition 5.5. ($\boldsymbol{CF_x}$ Explanations) Let x be a complete data instance and $f(x)$ its prediction by the decision function of f. A *counterfactual* explanation $\tilde{x}$ of x is such that:

i. $\tilde{x} \subseteq x$ ($\tilde{x}$ is a part of x)
ii. $f(x[\tilde{x}])$= 1-$f(x)$ (prediction inversion)
iii. There is no $\hat{x} \subset \tilde{x}$ such that $f(x[\hat{x}])$=$f(x[\tilde{x}])$ (minimality)

In definition 5.5, the term $x[\tilde{x}]$ denotes the data instance x where variables included in $\tilde{x}$ are inverted. In our approach, $\boldsymbol{CF_x}$ are enumerated thanks to the Minimal Correction Subset enumeration[10].

Definition 5.6. **(MSS)** A Maximal Satisfiable Subset (MSS) Φ of a CNF Σ is a subset (of clauses) $\Phi \subseteq \Sigma$ that is satisfiable and such that $\forall\ \alpha \in \Sigma \setminus \Phi$, $\Phi \cup \{\alpha\}$ is unsatisfiable.

Definition 5.7. **(MCS)** A Minimal Correction Subset Ψ of a CNF Σ is a set of formulas $\Psi \subseteq \Sigma$ whose complement in Σ, i.e., $\Sigma \setminus \Psi$, is a maximal satisfiable subset of Σ.

Following our modeling, an MCS for $\Sigma_f \cup \Sigma_x$ comes down to a subset of soft clauses denoted $\tilde{x}$, namely a part of x that is enough to remove (or reverse) in order to restore the consistency, hence to flip the prediction $f(x)$=0 to $f(x[\tilde{x}])$=1.

Proposition 5.8. *Let f be the decision function of the classifier, let Σ_f be its CNF representation. Let also x be a data instance predicted negatively ($f(x) = 0$) and $\Sigma_f \cup \Sigma_x$ the corresponding Partial Max-SAT encoding. Let $CF(x, f)$ be the set of counterfactuals of x wrt. f. Let $MCS(\Sigma_{f,x})$ the set of MCSs of $\Sigma_f \cup \Sigma_x$. Then:*

$$\forall \tilde{x} \subseteq x, \tilde{x} \in CF(x, f) \iff \tilde{x} \in MCS(\Sigma_{f,x}) \quad (3)$$

Proposition 5.8 states that each MCS of the CNF $\Sigma_f \cup \Sigma_x$ represents a CF $\tilde{x} \subseteq x$ for the prediction $f(x)$=0 and vice versa.

Example 5.9 (Example 5.4 cont'd). Given the CNF $\Sigma_f \cup \Sigma_x$ associated to RF_f from Example 4.3 and the input x=(1, 1, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 0, 1), we enumerate the counterfactual explanations to identify the minimal changes to alter the vote*Democrat* to *Republican*. There are four $\boldsymbol{CF_x}$:

- CF_x1="X_4=0 AND X_{12}=0" (meaning that in order to force the prediction to be 1, it is enough to alter x by setting only the variables *physician-fee-freeze* (X_4) and *education-spending* (X_{12}) to 1 while keeping the remaining values unchanged);
- CF_x2="X_5=0 AND X_{12}=0" ;
- CF_x3="X_5=0 AND X_9=1" ;
- CF_x4="X_4=0 AND X_5=0" ;

It is easy to see that the four $\boldsymbol{CF_x}$ allow to flip the negative prediction associated to x. Indeed, in Fig. 3, the pink lines show the branches of the trees that are fixed by the current input instance x. Clearly, according to CF_x1="X_4=0 AND X_{12}=0", if we set X_4=1 and X_{12}=1 then this will force DT_2 and DT_3 to predict 1 making the prediction of the random forest flip to 1.

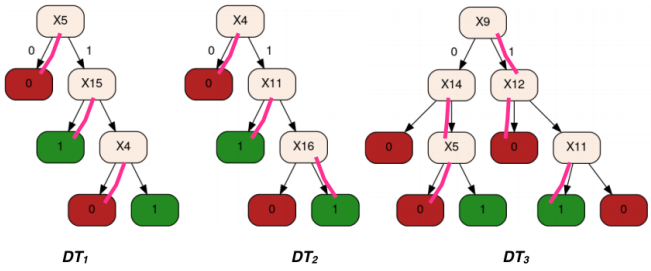


Figure 3: The random forest paths set by x

Until now, we presented **Step 1** allowing to encode in CNF a classifier f and **Step 2** allowing to enumerate symbolic explanations that are sufficient reasons and counterfactuals. There remains to numerically assess the relevance of such explanations on the one hand and assess the contribution to the prediction of each feature individually on the other hand. This is the objective of **Step 3** presented in the following section.

6 NUMERICALLY ASSESSING THE RELEVANCE OF SYMBOLIC EXPLANATIONS AND FEATURES

The number of symbolic explanations from **Step 2** can be large and a question then arises which explanations to choose or which explanations are most relevant?[6] We try to answer this question by defining some desired properties of an explanation score.
Hence, in order to select the most relevant[7] explanations and features, we propose to use some natural properties and propose some examples of scoring functions to assign a numerical score to an explanation and to a feature value of the input data.

6.1 Properties of symbolic explanations and scoring functions

Let us use $E(x, f)$ to denote the set of explanations (either $\boldsymbol{SR}_x$ or $\boldsymbol{CF}_x$) for an input instance x predicted negatively by the classifier f. An explanation is denoted by e_i where $i = 1, .., |E(x, f)|$ and $E(x, f)$ is a non empty set. The neighborhood of x within the radius r is formally defined as $V(x, r)$: $\{v \in X \mid \text{diff}(x, v) \leqslant r\}$[8]. Given an explanation e_i, let *size(e_i)* denote the number of variables composing it, and *Extent(e_i, x, r)* be the set of data instances defined as : $\{v \in V(x, r) \mid f(v)=f(x)$ and for $e_i \in E(v, f)\}$. Intuitively, *Extent(e_i, x, r)* denotes the set of data instances from the neighborhood of x that are negatively predicted by f and sharing the explanation e_i.
In the following we propose three natural properties that can be used to capture some aspects of our symbolic explanations :

- **Parsimony ($\mathcal{PAR}$)** : The parsimony is a natural property allowing to select the simplest or shortest explanations (namely, explanations involving less features). Hence, the parsimony score of an explanation e_i should be inversely proportional to it's size.
Formally, given a data instance x, its set of explanations $E(x, f)$: For two explanations e_1 and e_2 from $E(x, f)$: $\mathcal{PAR}(e_1) > \mathcal{PAR}(e_2)$ iff size(e_1)<size(e_2) . An example of a scoring function satisfying the parsimony property is :

$$S_{\mathcal{PAR}}(e_i) = \frac{1}{size(e_i)} \tag{4}$$

- **Generality ($\mathcal{GEN}$)** : This property aims to reflect how much an explanation can be general to a multitude of data instances, or in the opposite, reflect how much an explanation is specific to the instance. Intuitively, the generality of an explanation should be proportional to the number of data instances it explains. Given a data instance x, its set of explanations $E(x, f)$, its neighborhood $V(x, r)$ and two explanations e_1 and e_2 from $E(x, f)$: $\mathcal{GEN}(e_1) > \mathcal{GEN}(e_2)$ iff $|Extent(e_1, x, r)| > |Extent(e_2, x, r)|$. An example of a scoring function capturing this property is :

$$S_{\mathcal{GEN}}(x, r, e_i) = \frac{|Extent(e_i, x, r)|}{|V(x, r)|} \tag{5}$$

Intuitively, this scoring function assesses the proportion of data instances in the neighborhood of the instance x that are negatively predicted and that share the explanation e_i.

- **Explanation responsibility ($\mathcal{RESP}$)** : This property allows to answer the question how much an explanation is responsible for the current prediction. Intuitively, if there is a unique explanation, then this latter is fully responsible. Hence, the responsibility of an explanation should be inversely proportional to the number of explanations in $E(x, f)$. Given two different data instances x_1 and x_2 and their explanation sets $E(x_1, f)$ and $E(x_2, f)$ respectively and $e_k \in E(x_1, f) \cap E(x_2, f)$:
$\mathcal{RESP}(x_1, e_k) < \mathcal{RESP}(x_2, e_k)$ iff $|E(x_1, f)| > |E(x_2, f)|$. For a given data instance x, the responsibility of $e_i \in E(x, f)$ could be evaluated using the following scoring function :

$$S_{\mathcal{RESP}}(x, e_i) = \frac{1}{|E(x, f)|} \tag{6}$$

Note that the scoring function of Eq. 6 assigns the same score to every explanation in $E(x, f)$. To decide among the explanations in $E(x, f)$, one can calculate a responsibility score for e_i in the neighborhood of x. An example of a scoring function capturing this property, would be :

$$S_{\mathcal{RESP}}(x, r, e_i) = \max_{v \in V(x,r) \mid e_i \in E(v,f)} (S_{\mathcal{RESP}}(v, e_i)) \tag{7}$$

These properties make it possible to analyze and if necessary select or order the symbolic explanations according to a particular property. Of course, we can define other properties or variants of these properties (e.g. relative parsimony to reflect the parsimony of one explanation compared to the parsimony of the rest of the explanations). The properties can have a particular meaning or a usefulness depending on the applications and users. It would be interesting to study the links and the interdependence between these properties. Let us now see properties allowing to assess the relevance of the features reflecting their contribution to the prediction.

[6] An inconsistent Boolean formula can potentially have a large set of explanations (MUSes and MCSes). More precisely, for a knowledge base containing k clauses, the number of MUSes and MCSes can be in the worst case exponential in k [19].
[7] Of course, the relevance depends on the user's interpretation and the context.
[8] diff(x,v) denotes a distance measure that returns the number of different feature values between x and v.

	MNIST_0	MNIST_2	MNIST_5	MNIST_6	MNIST_8	SPECT	MONKS	Breast_cancer
avg acc of RF	98%	93%	99%	96%	95%	99%	98%	82%
min size CNF	1744/4944	1941/5452	2196/6102	1978/5534	1837/5178	2495/7174	2351/6714	5094/14184
avg size CNF	1979/5540	2172/6050	2481/6856	2270/6293	2059/5727	2758/7921	2883/8146	6069/16907
max size CNF	2176/6066	2429/6760	2789/7694	2558/7028	2330/6408	3088/8844	3451/9694	7053/19586
min enc_runtime (s)	0.83	0.88	0.92	0.82	0.74	1.07	1.66	2.02
avg enc_runtime (s)	1.05	1.06	1.11	0.92	0.86	1.214	1.56	2.5
max enc_runtime (s)	1.51	1.92	1.56	1.31	1.32	1.5	2.03	3.42

Table 1: Evaluating the scalability of the CNF encoding.

6.2 Properties of features-based explanations and scoring functions

Let us define *Cover(X_k,x)* as the set of explanations from $E(\mathrm{x}, f)$ where the feature X_k is involved (namely *Cover(X_k,x)*={e_i|$X_k \in e_i$ for $e_i \in E(x, f)$}). We consider the following properties :

- **Feature Involvement ($\mathcal{FI}$)** : This property is intended to reflect the extent of involvement of a feature within the set of explanations. The intuition is that a feature that participates in several explanations of the same instance x should have a higher importance compared to a less involved feature. Given a data instance x, its set of explanations $E(x, f)$, and two features X_1 and X_2: $\mathcal{FI}(X_1,x) > \mathcal{FI}(X_2,x)$ iff $|Cover(X_1,x)| > |Cover(X_2,x)|$. An example of a scoring function capturing this property is :

$$S_{\mathcal{FI}}(X_k, x) = \frac{|Cover(X_k, x)|}{|E(x, f)|} \tag{8}$$

- **Feature Generality ($\mathcal{FG}$)** : This property captures at what extent a feature is frequently involved in explaining instances in the vicinity of the sample to explain. Given a sample x, its vicinity $V(x, r)$ and the explanation set $E(V(x, r), f)$ defined as $\bigcup_{v \in V(x,r)} E(v, f)$, we have: $\mathcal{FG}(X_1) > \mathcal{FG}(X_2)$ iff $|\bigcup_{v \in V(x,r)} Cover(X_1, v)| > |\bigcup_{v \in V(x,r)} Cover(X_2, v)|$. An example of a scoring function capturing this property could be :

$$S_{\mathcal{FG}}(X_k) = \frac{|\bigcup_{v \in V(x,r)} Cover(X_k, v)|}{|E(V(x, r), f)|} \tag{9}$$

- **Feature Responsibility ($\mathcal{FR}$)** : This property is intended to reflect the responsibility or contribution of a feature X_i within the set of symbolic explanations of x. Intuitively, the responsibility of a feature should be inversely proportional to the size of the explanations where it is involved (the shortest the explanation, the highest the responsibility value of its variables). Given two features X_1, X_2 with non empty covers: $\mathcal{FR}(X_1) > \mathcal{FR}(X_2)$ iff $\underset{e_j \in Cover(X_1,x)}{aggr}(size(e_j)) < \underset{e_j \in Cover(X_2,x)}{aggr}(size(e_j))$ where *aggr* stands for an aggregation function (e.g. min, max, AVG, etc.). An example of a scoring function satisfying this property is :

$$S_{\mathcal{FR}}(X_k) = \frac{1}{\underset{e_j \in Cover(X_k,x)}{AVG}(size(e_j))} \tag{10}$$

Note that this is a non-exhaustive list of properties that one could be interested in order to select and rank explanations or features according to their contributions. In addition to the different explanation scores presented above, one can aggregate them (e.g., by averaging) to get an overall score depending on the user needs.

7 EMPIRICAL EVALUATION

7.1 Experimentation set-up

We evaluated our approach on a widely used standard ML dataset: the MNIST [9] handwritten digit database composed of 70,000 images of size 28 × 28 pixels. The images were binarized using a threshold $T = 127$. In addition, we used three other publicly available datasets (SPECT, MONKS and Breast-cancer). We trained *"one-vs-all"* binary neural network (BNN)[10] classifiers on the MNIST database to recognize digits (0 to 9) using the pytorch implementation[11] of the Binary-Backpropagation algorithm BinaryNets [13]. Neural network classifiers were trained on the rest of the datasets. Those classifiers are considered as the input black-box models we are interested in explaining their outcomes.

All experiments have been conducted on Intel Core i7-7700 (3.60GHz ×8) processors with 32Gb memory on Linux.

[9] http://yann.lecun.com/exdb/mnist/

7.2 Results

We report the following results by setting the following parameters $nb\_trees = 10$ and $max\_depth = 24$ for the random forest classifier trained on the vicinity of an input sample x as the surrogate model. The experiments were conducted on an average of 1500 instances picked randomly from the MNIST database. The predictions are made using [12] the "one-vs-all" BNN classifiers trained to recognize the 0,2,5,6 and 8 digits. Due to the limited number of pages, we only present the results for radius 250 with an average of 200 neighbors around x for MNIST. As for the rest, we consider all instances as neighbors (radius equal to the number of features).

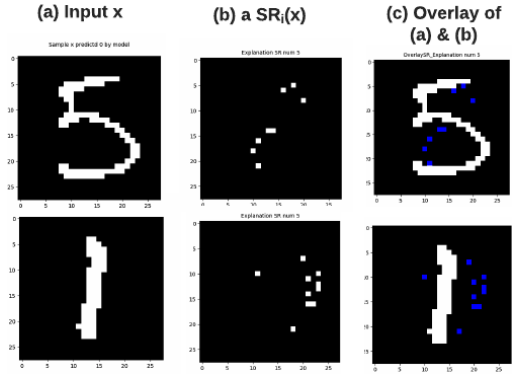
Evaluating the CNF encoding feasibility. We report our results regarding the size of the generated CNF formulae. We use the Tseitin Transformation [30] to encode the propositional formulae into an equisatisfiable CNF formulae.

Table 1 shows that the generated random forest classifiers provide interesting results in term of fidelity (high accuracy of the surrogate models) and tractability (size of the CNF encoding).In Table 1, the size of CNF is expressed as *number of variables/number clauses*. We can see that the number of variables and clauses of CNF formulae remains reasonable and easily handled by the current SAT-solvers which confirms the feasibility of the approach.

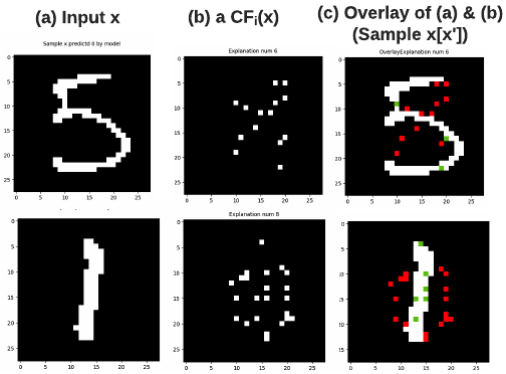
Evaluating the enumeration of symbolic explanations. The objective here is to assess the practical feasibility of the enumeration (scalability) of SR_x and CF_x explanations. For the enumeration of CF_x, we use the *EnumELSRMRCache tool*[13] implementing the boosting algorithm for MCSes enumeration proposed in [10] with a timeout

	MNIST_0	MNIST_2	MNIST_5	MNIST_6	MNIST_8	SPECT	MONKS	Breast_cancer
min #CFs	10	13	10	15	6	15	3	11
avg #CFs	35790	63916	99174	79520	4846	204	15	947
max #CFs	285219	546005	633416	640868	65554	700	41	5541
min enumtime (s)	0.005	0.11	0.006	0.11	0.008	0.01	0.01	0.02
avg enumtime (s)	21.49	42.11	77.72	50.86	2.35	0.12	0.03	1.5
max enumtime (s)	234.18	600	600	531.16	35.08	0.42	0.06	10.7

Table 2: Evaluating the enumeration of counterfactual explanations.



(a) Example of SR_x explanations.



(b) Example of CF_x explanations.

Figure 4: Data samples from MNIST database and their respective symbolic explanations.

set to 600s. As for the SR_x explanations, their enumeration can be easily done by exploiting the minimal hitting set duality relationship between MUSes and MCSes. Due to the page limitation, we only present the results about the enumeration of CF_x, but the results in terms of the number of explanations generated remain of the same order of magnitude.

[10] defined as a neural networks with binary weights and activations at run-time
[11] available at: https://github.com/itayhubara/BinaryNet.pytorch
[12] the results for the other digits are similar but can not be reported here because of space limitation
[13] available at http://www.cril.univ-artois.fr/enumcs/

We observe within **Table 2** that the average run-time remains reasonable (note that the times shown in Table 2 relate to the time taken to list all the explanations. The solver starts to find the first explanations very promptly) and that the approach is efficient in practice for medium size BNN classifiers (as shown in the experiments for BNNs with around 800 variables). We also observe that the number of CF_x may be challenging for a user to understand, hence the need for scoring them to filter them out and find the ones with the strongest influence on the prediction.

Illustrating SR_x and CF_x explanations for MNIST data set. We trained two "one-vs-all"[14] BNNs f_8 and f_0 to recognize the eight and zero digits. They have respectively achieved an accuracy of 97% and 99%. The "a" column in the different figures shows the input images (resp. representing the digit 5 and 1). Those data samples were negatively predicted. The model f_8 (resp. f_0) recognizes that the input image in the 1^{st} line (resp. the 2^{nd}) is not an 8-digit (resp. a 0-digit). Figure 4a shows an example of a single SR_x explanation highlighting the sufficient pixels for the models f_8 and f_0 to trigger a negative prediction. Figure 4b shows an example of CF_x explanations showing the pixels to invert in the input images to make the models f_8 and f_0 predict them positively. In addition, one could recognize in the "c" column of Fig. 4b a pattern of the 8-digit for the first image, and 0 for the second. It gives us a kind of "pattern/template" of the images that the model would positively predict.

Figure 5 shows heatmaps corresponding to the *Feature Involvement (FI)* scores (column "b-c") and *Feature Responsibility (FR)* (column "d-e") scores of the different input variables implicated in the SR_x and CF_x. Visually, they are simpler, clearer and easier to understand and use. We used around 100 data samples to compare the most important features according to the $\mathcal{FI}$ score of our approach and those of SHAP ("f" column of Fig. 5). The results coincide from 20% to 46% of cases, which is visually confirmed in our figures.

8 RELATED WORKS

Explaining machine learning systems has been a hot research topic recently. There has been hundreds of papers on ML explainability but we will be focusing on the ones closely related to our work. In the context of model-agnostic explainers where the learning function of the input model and its parameters are not known (black-box), we can cite some post-hoc explanations methods such as: LIME (Local Interpretable Model-agnostic Explanations) [23] which explain black-box classifiers by training an interpretable model g on samples randomly generated in the vicinity of the data instance. We follow an approach similar to LIME, the difference is that we encode our surrogate model into a CNF to generate symbolic explanations. The authors in [24] proposed a High-precision

[14] A "one-vs-all" BNN f_i returns a positive prediction for an input image representing the "i" digit, and negative one otherwise.

model agnostic explanations called ANCHOR. It is based on computing a decision rule linking the feature space to a certain outcome, and consider it as an anchor covering similar instances. Something similar is done in SHAP (SHapley Additive exPlanations) [20] that provides explanations in the form of the game theoretically optimal called Shapley values. Due to its computational complexity, other model-specific versions have been proposed for linear models and deep neural networks (resp LinearSHAP and DeepSHAP) in [20]. The main difference with this rule sets/feature-based explanation methods and the symbolic explanations we propose is that ours associates a score w.r.t to some relevance properties, in order to assess to what extent the measured entity is relevant as explanation or involved as features in the sufficient reasons or in the counterfactuals.

Recently, some authors propose symbolic and logic-based XAI approaches that can be used for different purposes [9]. We can distinguish the compilation-based approaches where Boolean decision functions of classifiers are compiled into some symbolic forms. For instance, in [8, 27] the authors showed how to compile the decision functions of naive Bayes classifiers into a symbolic representation, known as Ordered Decision Diagrams (ODDs). We proposed in a previous work [7] an approach designed to equip such symbolic approaches [27] with a module for counterfactual explainability. There are some ML models whose direct encoding into CNF is possible. For instance, the authors in [21] proposed a CNF encoding for Binarized Neural Networks (BNNs) for verification purposes. In [26], the authors propose a compilation algorithm of BNNs into tractable representations such as Ordered Binary Decision Diagrams (OBDDs) and Sentential Decision Diagrams (SDDs). The authors in [28] proposed algorithms for compiling Naive and Latent-Tree Bayesian network classifiers into decision graphs. In [2], the authors dealt with a set of explanation queries and their computational complexity once classifiers are represented with compiled representations. However, the compilation-based approaches are hardly applicable to large sized models, and remain strongly dependent on the type of classifier to explain (non agnostic). Our approach can use those compilation algorithms to represent the whole classifier when the encoding remains tractable, but in addition, we propose a local approximation of the original model using a surrogate model built on the neighborhood of the instance at hand.

Recent works in [16, 17] deal with some forms of symbolic explanations referred to as abductive explanations (AXp) and contrastive explanations (CXp) using SMT oracles. In [14], the authors explain the prediction of decision list classifiers using a SAT-based approach. Explaining random forests and decision trees is dealt with for instance in [2] and [15, 18] respectively. The main difference with our work, is that we are proposing an approach that goes from the model whose predictions are to be explained to its encoding and goes beyond the enumeration of symbolic explanations by defining some scoring functions w.r.t some relevance properties. Different explanation scores have been proposed in the literature. Authors in [4] used the counterfactual explanations to define an explanation responsibility score for a feature value in the input. In [5], the authors used the answer-set programming to analyze and reason about diverse alternative counterfactuals and to investigate the causal explanations and the responsibility score in databases.

9 CONCLUDING REMARKS AND DISCUSSIONS

We proposed a novel model agnostic generic approach to explain individual outcomes by providing two complementary types of symbolic explanations (*sufficient reasons* and *counterfactuals*) and scores-based ones. The objective of the approach is to explain the predictions of a black-box model by providing both symbolic and score-based explanations with the help of Boolean satisfiability concepts. The approach takes advantage of the strengths of already existing and proven solutions, and of the powerful practical tools for the generation of MCS/MUS. The proposed approach overcomes the complexity of encoding a ML classifier into an equivalent logical representation by means of a surrogate model to symbolically approximate the original model in the vicinity of the sample of interest. The presentation of the paper was limited to the explanation of negative predictions to exploit the concepts of MUS and MCS and use a SAT-based approach. For positively predicted instances,

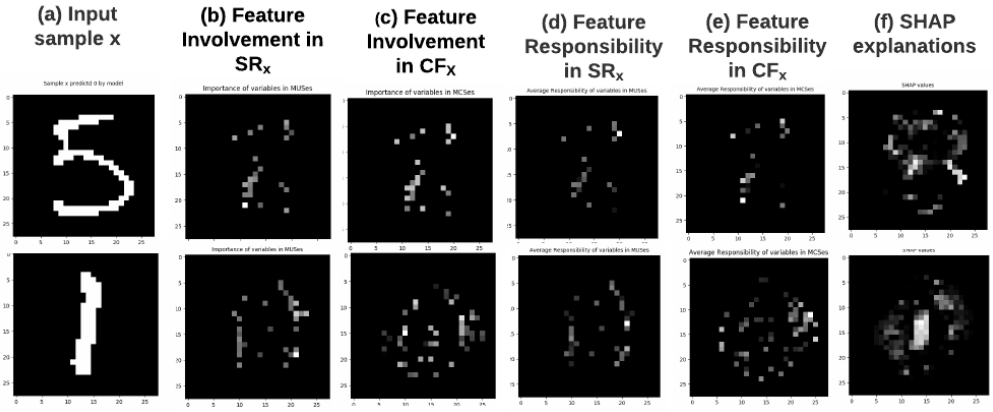


Figure 5: Heatmaps in columns (b-c) representing the (FI) score, and (d-e) the (FR) computed over the SR_x and CF_x of the samples data from MNIST (column a) in comparaison to heatmaps of the SHAP values (column f).

we can simply work on the negation of the symbolic representation (CNF) of f (namely $\neg\Sigma_f$). The enumeration of the explanations is done in the same way as for negative predictions.
To the best of our knowledge, our approach is the first that generates different types of symbolic explanations and **fine-grained** score-based ones. In addition, our approach is **agnostic** and **declarative**. Another advantage of our approach is the **local faithfulness** [23] to the instance to be explained.As future works, we intend to extend our approach for multi-label (ML) classification tasks to explain predictions in a multi-label setting.

ACKNOWLEDGMENTS

The authors would like to thank the Région Hauts-de-France and the University of Artois for supporting this work.

REFERENCES

[1] Ignasi Abío, Robert Nieuwenhuis, Albert Oliveras, and Enric Rodríguez-Carbonell. 2013. A parametric approach for smaller and better encodings of cardinality constraints. In *International Conference on Principles and Practice of Constraint Programming*. Springer, 80–96.

[2] Gilles Audemard, Frédéric Koriche, and Pierre Marquis. 2020. On Tractable XAI Queries based on Compiled Representations. In *Proceedings of the 17th International Conference on Principles of Knowledge Representation and Reasoning*. 838–849. https://doi.org/10.24963/kr.2020/86

[3] Olivier Bailleux and Yacine Boufkhad. 2003. Efficient CNF encoding of boolean cardinality constraints. In *International conference on principles and practice of constraint programming*. Springer, 108–122.

[4] Leopoldo E. Bertossi. 2020. Declarative Approaches to Counterfactual Explanations for Classification. *CoRR* abs/2011.07423 (2020). arXiv:2011.07423 https://arxiv.org/abs/2011.07423

[5] Leopoldo E. Bertossi. 2020. Score-Based Explanations in Data Management and Machine Learning. In *Scalable Uncertainty Management - 14th International Conference, SUM 2020, Bozen-Bolzano, Italy, September 23-25, 2020, Proceedings (Lecture Notes in Computer Science, Vol. 12322)*, Jesse Davis and Karim Tabia (Eds.). Springer, 17–31. https://doi.org/10.1007/978-3-030-58449-8_2

[6] Armin Biere, Marijn Heule, and Hans van Maaren. 2009. *Handbook of satisfiability*. Vol. 185. IOS press.

[7] Ryma Boumazouza, Fahima Cheikh-Alili, Bertrand Mazure, and Karim Tabia. 2020. A Symbolic Approach for Counterfactual Explanations. In *International Conference on Scalable Uncertainty Management*. Springer, 270–277.

[8] H. Chan and Adnan Darwiche. 2003. Reasoning about Bayesian Network Classifiers. In *UAI*.

[9] Adnan Darwiche. 2020. Three Modern Roles for Logic in AI. In *Proceedings of the 39th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems* (Portland, OR, USA) *(PODS'20)*. Association for Computing Machinery, New York, NY, USA, 229–243. https://doi.org/10.1145/3375395.3389131

[10] Éric Grégoire, Yacine Izza, and Jean-Marie Lagniez. 2018. Boosting MCSes Enumeration.. In *IJCAI*. 1309–1315.

[11] Eric Grégoire, Bertrand Mazure, and Cédric Piette. 2007. Boosting a Complete Technique to Find MSS and MUS Thanks to a Local Search Oracle.. In *IJCAI-07*, Vol. 7. 2300–2305.

[12] Tin Kam Ho. 1995. Random decision forests. In *Proceedings of 3rd international conference on document analysis and recognition*, Vol. 1. IEEE, 278–282.

[13] Itay Hubara, Matthieu Courbariaux, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. 2016. Binarized Neural Networks. In *Advances in Neural Information Processing Systems*, Vol. 29.

[14] Alexey Ignatiev and Joao Marques-Silva. 2021. SAT-Based Rigorous Explanations for Decision Lists. In *Theory and Applications of Satisfiability Testing – SAT 2021*, Chu-Min Li and Felip Manyà (Eds.). Springer International Publishing, Cham, 251–269.

[15] Alexey Ignatiev, Nina Narodytska, Nicholas Asher, and João Marques-Silva. 2020. On Relating 'Why?' and 'Why Not?' Explanations. *CoRR* abs/2012.11067 (2020). arXiv:2012.11067 https://arxiv.org/abs/2012.11067

[16] Alexey Ignatiev, Nina Narodytska, and Joao Marques-Silva. 2019. Abduction-based explanations for machine learning models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 1511–1519.

[17] Alexey Ignatiev, Nina Narodytska, and Joao Marques-Silva. 2019. On Relating Explanations and Adversarial Examples. In *Advances in Neural Information Processing Systems*, Vol. 32.

[18] Yacine Izza, Alexey Ignatiev, and João Marques-Silva. 2020. On Explaining Decision Trees. *CoRR* abs/2010.11034 (2020). arXiv:2010.11034 https://arxiv.org/abs/2010.11034

[19] Mark H Liffiton and Karem A Sakallah. 2008. Algorithms for computing minimal unsatisfiable subsets of constraints. *Journal of Automated Reasoning* 40, 1 (2008), 1–33.

[20] Scott M Lundberg and Su-In Lee. 2017. A Unified Approach to Interpreting Model Predictions. In *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.), Vol. 30. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2017/file/8a20a8621978632d76c43dfd28b67767-Paper.pdf

[21] Nina Narodytska, Shiva Kasiviswanathan, Leonid Ryzhyk, Mooly Sagiv, and Toby Walsh. 2018. Verifying properties of binarized deep neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 32.

[22] Raymond Reiter. 1987. A theory of diagnosis from first principles. *Artificial intelligence* 32, 1 (1987), 57–95.

[23] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. " Why should i trust you?" Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. 1135–1144.

[24] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2018. Anchors: High-precision model-agnostic explanations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 32.

[25] Ron Rymon. 1994. An se-tree-based prime implicant generation algorithm. *Annals of Mathematics and Artificial Intelligence* 11, 1 (1994), 351–365.

[26] Weijia Shi, Andy Shih, Adnan Darwiche, and Arthur Choi. 2020. On Tractable Representations of Binary Neural Networks. In *Proceedings of the 17th International Conference on Principles of Knowledge Representation and Reasoning, KR 2020, Rhodes, Greece, September 12-18, 2020*, Diego Calvanese, Esra Erdem, and Michael Thielscher (Eds.). 882–892. https://doi.org/10.24963/kr.2020/91

[27] Andy Shih, Arthur Choi, and Adnan Darwiche. 2018. A Symbolic Approach to Explaining Bayesian Network Classifiers. In *IJCAI-18*. International Joint Conferences on Artificial Intelligence Organization, 5103–5111. https://doi.org/10.24963/ijcai.2018/708

[28] Andy Shih, Arthur Choi, and Adnan Darwiche. 2019. Compiling Bayesian Network Classifiers into Decision Graphs. In *Proceedings of the AAAI-19*, Vol. 33. 7966–7974.

[29] Carsten Sinz. 2005. Towards an optimal CNF encoding of boolean cardinality constraints. In *International conference on principles and practice of constraint programming*. Springer, 827–831.

[30] Grigori S Tseitin. 1983. On the complexity of derivation in propositional calculus. In *Automation of reasoning*. Springer, 466–483.