TT-RULES: EXTRACTING & OPTIMIZING EXACT RULES OF A CNN-BASED MODEL - APPLICATION TO FAIRNESS

Anonymous authors
Paper under double-blind review

ABSTRACT

Most Machine Learning (ML) models are “black box” models, but in critical domains such as healthcare, energy, finance, military, or justice, they need to be globally and exactly interpretable. Creating ML models convertible by design into rule-based models is an attractive solution: they produce all the rules (global nature of interpretability) that allow us to obtain exactly the output result (exact nature of interpretability). Today, these rule-based models are mainly decision trees, whose natural interpretability is outweighed by their poor performances and scalability. In this paper, we offer a new three-step framework, TT-rules, that extracts and optimizes exact rules from a recent family of Convolution Neural Networks (CNNs) called Truth Table nets (TTnets). First, we show how to extract rules $\mathcal{R}$ in Disjunction Normal Form (DNF) from TTnets, which we adapt and enhance for tabular datasets. Secondly, we explain how the TT-rules framework permits the optimization of two key interpretability factors, namely the number of rules and their size, transforming the original set $\mathcal{R}$ into an optimized $\mathcal{R}_{opt}$. Our rule-based model is thus composed of $\mathcal{R}_{opt}$ with a final binary linear regression and allows multi-label classification. In a third step, we improve the rules’ visualization by converting them into Reduced Ordered Binary Decision Diagrams (ROBDD) and enriching them by computing interesting associated probabilities. To evaluate TT-rules’ performances, we applied it to two tabular healthcare datasets and two fairness datasets. Our framework reaches competitive results compared to state-of-the-art rule-based models in terms of accuracy, complexity, and statistical parity, also giving exact and global interpretability. In addition, we show that practitioners can use their domain knowledge to diagnose individual fairness of a given TT-rules model by analyzing and further modifying the rules $\mathcal{R}_{opt}$. As an example of the compactness of our framework’s output, we draw all the rules in $\mathcal{R}_{opt}$ for one model on the Adult dataset (only 15 conditions for an 84.6% accuracy).

1 INTRODUCTION

Lack of explainability and difficulty in integrating human knowledge are well-known concerns for Deep Neural Networks (DNNs) and more generally for all ensemble ML models due to their large complexity Zhang et al. (2021); He et al. (2020); Ribeiro et al. (2016; 2018); Molnar (2020); Fryer et al. (2021). Therefore, the global and exact interpretability of these systems is the subject of intense research efforts, especially for safety-critical applications Driscoll (2020); Regulation (2016); Osoba & Welser IV (2017).

In practice, rule-based models Freitas (2014), like tree-based models Quinlan (1986; 1987); Bessiere et al. (2009), can easily provide global explanations, i.e. independent of the input, and exact, i.e. giving the same output as the model. However, despite their intrinsic interpretable nature, their performances are lower than those of other less interpretable family models including DNN or ensemble ML models such as Random Forest Breiman (2001).

Rule-based models produce predicates for the decision that are expressed in DNF. For example, in the Adult Census dataset Dua & Graff (2017) one rule of the model for determining whether an individual would be reaching yearly earnings exceeding 50K$/year might be:

$$((\text{Age} > 34) \wedge \text{Maried}) \vee (\text{Male} \wedge (\text{Capital Loss} < 1\text{k/year})) \vee ((\text{Age} < 34) \wedge \text{Go to University})$$

Despite the large number of works published on DNN interpretability Zhang et al. (2021); He et al. (2020), few can provide as strong interpretations as those from rule-based models. Rules are particularly suited for tabular data that contain mixed-type features and exhibit complex high-order feature interactions. The quality assessment of a model is usually based on three criteria: a) the performance, b) the number of rules and c) the size of each of the rules Freitas (2014).

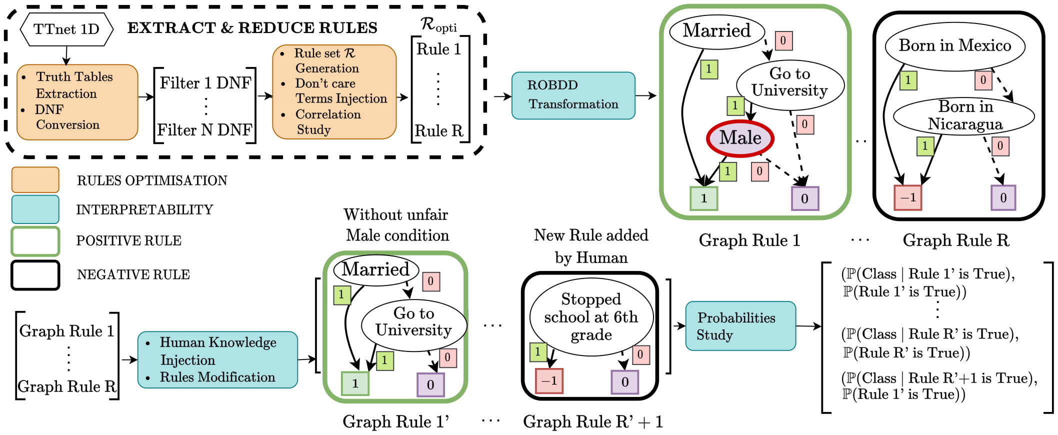
In Benamira et al. (2022), the authors present a new convolutional neural network that is, by design, convertible into SAT formulas. This is achieved through first lowering the number of connections from one Deep Convolutional Neural Network layer to another, and secondly by binarising the input/output of the filters. Thus, they decrease the complexity of the CNNs' filters and facilitate their encoding into SAT formulas, while maintaining the model accuracy for image classification tasks. This novel neural network was named Truth Table Deep Convolutional Neural Network (TTnet) as it uses internally small lookup tables. For clarity's sake, Figure 4 in Appendix A reproduces the illustration of the architecture of the model and of its key component, the Learning Truth Table (LTT) block. TTnet was introduced for formal verification applications on image datasets (MNIST and CIFAR-10), and we propose here an application to rule-based models.

Our contributions. In this paper, we present TT-rules, a three-step framework that constitutes, to the best of our knowledge, the first CNN rule-based model for multi-label classification. **I)** First, in order to apply TTnet family Benamira et al. (2022), which originally uses two-Dimensional-Convolutional Neural Network (2D-CNN), on a tabular dataset, we extend it to 1D-CNN. To overcome the natural limitation of TTnet being decoupled, we introduce a new input redundancy strategy to increase the accuracy performance. Then **II)**, leveraging the technique proposed in the original TTnet paper, we transform each of the filters into a truth table, followed by a conversion into a DNF equation. Another contribution of our work is the addition of a final process to convert each DNF equation into a set of rules $\mathcal{R}$. Next, to enhance this transformation, we propose two optimizations to make the post-processing interpretability readily amenable by experts. Indeed **II -a)** first we decrease the size of each rule in $\mathcal{R}$ by incorporating the *don't care terms* in truth tables thanks to human logic, an operation not presented in the original paper Benamira et al. (2022). **II -b)** Secondly, we decrease the number of rules in $\mathcal{R}$ by introducing the inter-rules correlations score and by analyzing them. These two optimizations, only possible for TTnet, transform the set $\mathcal{R}$ into an optimized one $\mathcal{R}_{opt}$. We clearly quantify the trade-off between performance, the number of rules, and the rules' size. At the end of the second step, the rule-based model is created. For class prediction, one only needs to sum all the rules in $\mathcal{R}_{opt}$ according to a binary linear regression. **III)** In a third step, we further enhance the global and exact interpretability **III -a)** by first converting all rules equations to ROBDD for better visualization. **III -b)** Then, we propose to compute valuable probabilities associated with each rule of the set of rules $\mathcal{R}_{opt}$. For instance, the probability to have a class $\texttt{C}$ knowing that the rule $\texttt{r} \in \mathcal{R}_{opt}$ is activated $\mathbb{P}(\texttt{C} \mid \texttt{r is True})$ and others relevant properties.

To illustrate the interpretable value of our framework, we apply it to two critical fields: healthcare (Cancer and Diabetes datasets) and fairness (Adult and Compas datasets). We demonstrate that our framework reaches competitive results when compared to state-of-the-art rule-based models in terms of accuracy, complexity, and group-fairness measured with statistical parity while giving exact and global explanations. In addition, applying our approach to the field of individual-fairness, we establish that human knowledge-based post-processing allows us to improve the model through a clear understanding of the learned rules and a compact decision-making mechanism. We show that given a sensitive feature and a rule set $\mathcal{R}_{opt}$, "ghost proxies rules" can be manually identified and targeted. They are rules that are difficult to discover in models (hence *ghost*) and that contain features that are proxies of the sensitive attribute (hence *proxies*). We precisely illustrate the above statement by developing a concrete TT-rules model use-case study on the Adult dataset, given in Figure 2. In addition, we managed to generate 5 extremely compact rules of small sizes (15 conditions in total) while retaining a good accuracy of 84.6% on Adult dataset (for comparison other rule-set-models Dash et al. (2018); Cohen & Singer (1999); Angelino et al. (2017) provide 82.3% accuracy for 4 conditions or 84.9% for 230 conditions on the same dataset). The global TT-rules framework is depicted in Figure 1.

Outline. Sections 2 describe related works. Section 3 presents our three-step TT-rules framework. Section 4 details the TT-rules experiments on healthcare and fairness datasets. Finally, we present TT-rules limitations and conclude in Section 5.

Figure 1: TT-rules general framework. Top process represents the steps **I** to **III-a)**. In the bottom process, the Male attribute has been removed from for Graph Rule 1 by the human practitioner to create Graph Rule 1'. Graph Rule R'+1 has also been added by the practitioner to improve performance. Probabilities considered during step **III-b)** can be computed on each $\mathcal{R}$ and $\mathcal{R}_{\text{opt}}$ sets.



2 Related Work

Rule-based models. Along decision trees Quinlan (1986; 1987); Bessiere et al. (2009), rules lists Rivest (1987); Angelino et al. (2017); Dash et al. (2018) or bayesian networks Friedman et al. (1997); Freitas (2014), rules sets Lakkaraju et al. (2016); Cohen (1995); Cohen & Singer (1999); Quinlan (2014); Wei et al. (2019) are part of the rule-based models family. In rules sets, each rule encodes a different class and all have the same weight. Therefore, our work may be classified in the rules sets field. Interestingly, in Lakkaraju et al. (2016), the authors show that rules sets are ultimately more interpretable than rules lists as they are easier to infer. Yet, to the best of our knowledge, none of the previous rules sets models managed to propose rules for more complex tasks than tabular datasets classification, like image datasets classification, for example Yang et al. (2021); Wang et al. (2021). Moreover, most of them handle only binary classification. We choose to continue the work of Benamira et al. (2022) as the authors proposed a novel CNN architecture that is encodable by construction into DNF formulas Biere et al. (2009), that can scale to image datasets (MNIST and CIFAR10) and can be extended to multi-label classification.

Fairness study. The recognition of dataset biases highlights the limits of learning generalization Tommasi et al. (2017); Torralba & Efros (2011). As everyday life decision-making increasingly relies on data-driven ML systems, the need for fairness-guaranteed algorithms arose in the regulative arena Commission (2021). Fairness can be enforced on different levels: on the individual level, i.e. giving similar treatment to similar individuals when only the sensitive attribute changed McNamara et al. (2019); Dwork et al. (2012), and on a group level, i.e. not discriminating against a subgroup such as a specific gender or ethnicity Dwork et al. (2012); Hardt et al. (2016); Balunovic et al. (2021); Zafar et al. (2019). Fairness can be forced at different phases of the ML process: directly on the dataset Kamiran & Calders (2012), as a training constraint Du et al. (2021); Kamishima et al. (2012) or as post-processing Hardt et al. (2016); Pleiss et al. (2017). We refer to Mehrabi et al. (2021) for an extensive study of related works on the topic. In this paper, we focus on individual and group fairness at the post-processing level, exploiting the global and exact interpretability property of our model.

3 TT-rules framework

In this section, we detail our three-step TT-rules framework: **I)** the model innovations induced by the new nature of the application, i.e. tabular dataset classification (Section 3.1); **II)** the generation of the set of rules $\mathcal{R}$ and its optimized version $\mathcal{R}_{opt}$ (Sections 3.2.1 and 3.2.2) as well as a formal definition of our rule-based model (Section 3.2.3); **III)** the ROBDD rule format and the Rules Distribution Table computation to enhance the rule interpretability (Sections 3.3.1 and 3.3.2).

3.1 Step 1: model training

From 2D-CNN to 1D-CNN. TTnet was originally designed for image classification applications. However, global and exact interpretability is easier on tabular datasets. Hence, we are adjusting some of the elements of the original TTnet model as follows. First, we need to convert the 2D-LTT blocks into 1D-LTT blocks. To achieve this, the normalized input of size (B, F), with $B =$"Batch" and $F =$ "Number of Features" is converted into $(B, 1, F)$: which corresponds to a time series with 1 feature and F time steps. Then, we need to change the 2D-CNN autoencoder of the LTT block into a 1D-CNN autoencoder. The rest of the model remains unchanged. Notably, the final layer is a sparse binary classification: it has double advantage of reducing the number of rules and considering that all rules have the same weight.

Permutation strategy: improvement of the accuracy. The TTnet decoupled nature is at the same time its strength and its weakness. It means that one feature of V, the final vector before the final binary linear regression, can only see a subpart of all the inputs - whereas in MLP one feature of V can see all the inputs. To overcome this natural drawback of TTnet, we present a straightforward and efficient strategy that randomly tries multiple pairs of features. More precisely, the updated dataset $\mathcal{D}_f$ is composed of the original dataset $\mathcal{D}$, concatenated will P permutated versions of its features: $\mathcal{D}_f = \mathcal{D}||Perm_1(\mathcal{D})||\dots||Perm_P(\mathcal{D})$, where $||$ denotes the concatenation operation and $Perm_i$ are randomly chosen permutations on F elements. We demonstrate in Table 3 in Appendix B that this method improves the TTnet accuracy.

3.2 Step 2: Conversion from TTnet to TT-rules

3.2.1 From Truth Table to set of Rules

General. Once the TTnet is trained, we need to extract the exact rules set $\mathcal{R}$. To do so, we first need to generate the truth table for the LTT blocks. As detailed previously in Benamira et al. (2022), we generate all the inputs possible, pass them as LTT block input, and then we save its output: resulting in the truth table. Then, we convert the truth table to general DNF form with the Quine-McCluskey algorithm Quine (1952). Finally, we introduce a method to convert the general DNF form into rules. We simply divide the input into patches and we replace the DNF literal with the corresponding feature name: the number of rules for one filter corresponds to the number of patches. We repeat the operation for all LTT blocks to get $\mathcal{R}$. An example is given below.

Example 1. Consider the following 6-feature binary input: [Age > 34, Male, Go to University, Married, Born in US, Born in France]. In this example the permutation strategy is not used. We consider a 1D-CNN layer with one filter, a kernel size of 4, a stride of size 2, and no padding. The 1D-CNN weights are: $W_1 = [10\ \ \text{-1}\ \ 3\ \ \text{-5}]$. We want to highlight here that for simplicity's sake, we consider filters and not LTT blocks. We define the following DNF literals $F = [x_0\ \ x_1\ \ x_2\ \ x_3]$. As the inputs and output are binary and as the number of entries of the CNN is 4 (kernel size = 4), we have $2^4 = 16$ possible entries: [0 0 0 0], [0 0 0 1], $\cdots$, [1 1 1 1]. For each input, we calculate the corresponding output resulting from the convolution of W_1 with the 16 possible literal entries: [0, -5, 3, -2, -1, -5, 3, -2, 10, 5, 13, 8, 9, 4, 12, 7]. After binarization with the Heaviside step function (i.e. $Bin(x) = \frac{1}{2} + \frac{sgn(x)}{2}$), we have $y_{binary} =$ [0, 0, 1, 0, 0, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1]. We then apply the Quine-McCluskey algorithm Quine (1952) which gives after simplification $\mathsf{SAT}_1^{\mathsf{DNF}} = (x_2 \wedge \overline{x_3}) \vee x_0$. This procedure provides us the general form of the filter but not the rules. To get the rules, we need to consider the padding and the stride of the LTT block. In our case, with a stride at 2 and no padding, we get 2 patches: Age$>$ 34, Male, Go to University, Married] and [Go to University, Married, Born in the US, Born in France]. After replacement of the literal by the corresponding feature, we get 2 rules: $\mathsf{Rule}_1^{\mathsf{DNF}} = (\text{Go to University} \wedge \overline{\text{Married}}) \vee (\text{Age} > 34)$ and $\mathsf{Rule}_2^{\mathsf{DNF}} = (\text{Born in the US} \wedge \overline{\text{Born in France}}) \vee$ Go to University. Because of the decoupled nature of TTnet, in such configuration the feature "Born in France" will never be combined as it is with the feature "Age$>$ 34" and the permutation strategy was proposed to overcome this limitation. We also underline the logic redundancy in $\mathsf{Rule}_2^{\mathsf{DNF}}$ for the place of birth: if someone is born in the US, he/she is necessarily not born in France. The next subsection solves this issue.

3.2.2 OPTIMIZATIONS

The optimizations presented in this section are uniquely applicable for the TTnet model as both require that a CNN filter is expressed as a truth table.

Don't care terms **(DCT) injection: reducing the rule's size.** We propose to inject *don't care terms* (DCT) into the truth table. They represent the fact that for a particular input, the LTT block output value can be equivalently 0 or 1 and the overall performance of the DNN will not be affected. The Quine-McCluskey algorithm will then assign the optimal value to the DCT to reduce the DNF equations. Here the injection of DCT is achieved through human common sense and reasoning and the above Example 1 nicely illustrates this: no one can be born in France and the US at the same time. Therefore, this implies that the literals x_2 and x_3 must not be at 1 at the same time, uniquely for the second rule. Then, the DNF equation changes: we inject the DCT inside the truth table as y_{binary} $=$ [0, 0, 1, DCT, 0, 0, 1, DCT, 1, 1, 1, DCT, 1, 1, 1, DCT]. We thus get the new rule: $\mathsf{Rule}^{\mathsf{DNF}}_{2,reduced}$ $=$ Born in the US $\vee$ Go to University. This drastically decreases the rules' size while maintaining the accuracy unchanged, see Table 3 in Appendix B for more detailed results.

Truth Table Correlation metric: reducing the number of rules. To reduce the number of rules obtained with TT-rules, we introduce a new metric called Truth Table Correlation (TTC). This metric is based on the idea that two different LTT blocks may learn similar rules as they are all completely decoupled from each other. Thus, to prevent rules redundancy, we introduce a correlation measure between two LTT block rules. It is defined as follows:

$$TTC(y_1, y_2) = \begin{cases} \frac{HW(y_1,\overline{y_2})}{|y_1|} - 1 & \text{if } abs(\frac{HW(y_1,\overline{y_2})}{|y_1|} - 1) > \frac{HW(y_1,\overline{y_2})}{|y_1|} \\ \frac{HW(y_1,y_2)}{|y_1|} & \text{otherwise.} \end{cases}$$

where y_1 and y_2 are the outputs of the LTT blocks, $\overline{y_2}$ is the negation of y_2, $|y_1|$ represents the number of elements in y_1, HW is the Hamming distance function (the Hamming distance between two equal-length strings of symbols is the number of positions at which the corresponding symbols are not equal). TTC metric varies from -1 to 1. For $TTC = -1$, the LTT blocks are exactly opposite while they are the same if $TTC = 1$. We systematically filter redundant rules above the threshold correlation of ± 0.9. If the correlation is positive, we delete one of the two filters and give the same value to the second filter. If the correlation is negative, we delete one of the two filters and give the opposite value to the second filter.

3.2.3 RULE-BASE MODEL

At this step, we can define the optimized rules set $\mathcal{R}_{opt}$, generated from $\mathcal{R}$ by applying both strategies of DCT injection and filtering rules with high Truth Table Correlation. The rules in $\mathcal{R}_{opt}$ are eventually summed up according to the final binary linear regression. As the final binary linear regression is very sparse (more than 90%), it means that most rules do not take part in the prediction class: we will only keep the ones that are summed up. Let's consider a binary classification and the optimized set of rules $\mathcal{R}_{opt}$. It can be divided into $\mathcal{R}_{opt,+}$ the set of rules that are used for class 1 in the final binary linear regression, and $\mathcal{R}_{opt,-}$ the set of rules that are used for class 0. Then, considering an input I, and the set of rules $\mathcal{R}_{opt} = \mathcal{R}_{opt,+} \cup \mathcal{R}_{opt,-}$ the rule-based model can be defined as follows:

$$\text{Classifier(I, } \mathcal{R}_{opt}\text{)} = \begin{cases} 1 & \text{if } \sum_{r_+ \in \mathcal{R}_{opt,+}} \mathbb{1}_{\{r_+(I) \texttt{ is True}\}} - \sum_{r_- \in \mathcal{R}_{opt,-}} \mathbb{1}_{\{r_-(I) \texttt{ is True}\}} > 0 \\ 0 & \text{otherwise.} \end{cases}$$

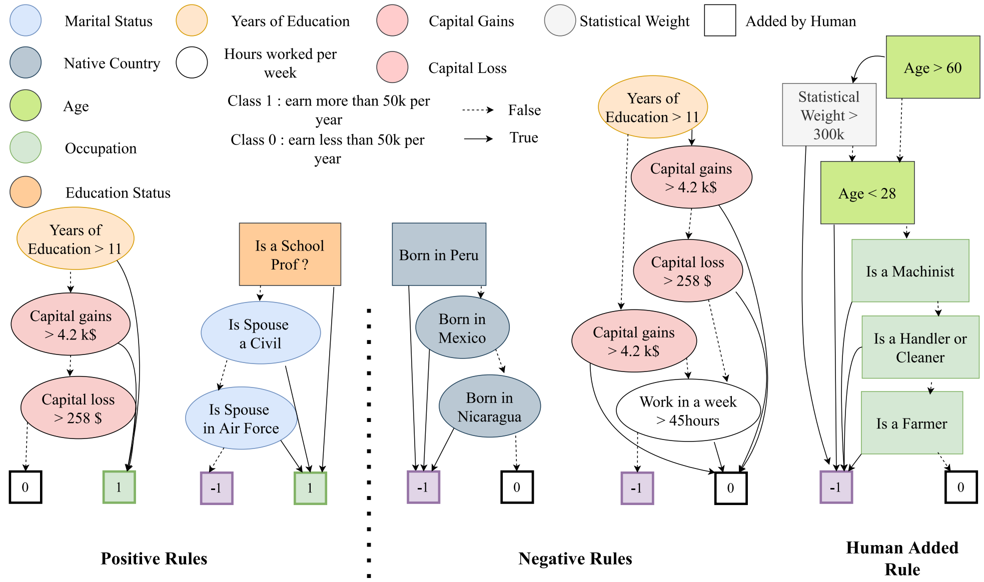
with "$r(I)$ `is True`" meaning that the input I fulfilled the rule r. In Example 1, it can be that I is born in the US to fulfill the rule $Rule^{DNF}_{2,reduced}$. This model can be extended to multi-label classification, like all final linear classification matrices.

3.3 STEP 3: ENHANCING THE INTERPRETABILITY

3.3.1 ROBDD RULE FORMAT

At this step, a rule is in DNF format. However, we prefer to have the rules in a decision tree format. Therefore, we transform our DNF into its equivalent ROBDD graph. A Binary Decision Diagram (BDD) is a directed acyclic graph used to represent a Boolean function. They were originally introduced by Lee (1959); Akers (1978). In Bryant (1986), Randal Bryant introduced the Reduced Ordered BDD (ROBDD), which is a canonical form given an identical ordering of input variables, equivalent Boolean functions will always reduce to the same ROBDD. This is a desirable property for determining formal equivalence and very useful for visualizing rules. We perform the transformation with PyEda library Aqajari et al. (2021). Graphs, given in Figure 2, are small and easy to understand.

Figure 2: Adult dataset case study: our rule-based model with $\mathcal{R}_{opt}$. Added conditions are represented in rectangles. The original model reaches 83.6% accuracy. By modifying existing rules and incorporating the **Human Added Rule** we reach 84.6% accuracy. On the same test set, Random Forest reaches 85.1% accuracy and Decision Tree 84.4% with depth 10. Table 1 gives related probabilities. Rules 1 to 5 are numbered from left to right. Rule 2 was negatively correlated with another rule, and as described in Section 3.2.2, the other is deleted and the -1 weight is related to a non activation of the Rule 2. This shows that we can to add human knowledge to a CNN-based model. Additional results can be found in Appendix C.



3.3.2 Rules Probability computation

Motivations. At this stage, there are still two main limits to interpretability for experts. First, in the case where the set $\mathcal{R}_{opt}$ is large, the ranking rules problem emerges. Practitioners need to know which rules have more impact than others. Besides, if a user only knows if one rule is activated, then the model does not give the user any information on how close he is getting the prediction class 0 for example. More specifically, in Example 1, if the rule $\mathsf{Rule}^{\mathsf{DNF}}_{2,reduced} =$ Born in the US $\vee$ Go to University is fulfilled for an individual, it only gives us the information that one of the rules goes in the direction of "win more than 50K$/per year". We would prefer to know what is the probability that the class prediction is 1, without computing the other rules, and also how often this rule is true.

Offline Computation of the Rules Distribution Table (RDT). Once the optimal ensemble $\mathcal{R}_{opt} = \mathcal{R}_{opt,+} \cup \mathcal{R}_{opt,-}$ of relevant rules is determined, we compute the Rules Distribution Table RDT: for all rules $r \in \mathcal{R}_{opt}$, we compute and store $\mathbb{P}(\texttt{Class|\ r is True}) = \mathbb{P}(\texttt{C|\ r is True})$ and $\mathbb{P}(\texttt{r is True})$. Algorithm 1 in Appendix D describes our construction method. The inputs of the algorithm are the training dataset and $\mathcal{R}_{opt}$. The output RDT is constructed as follows: first, we compute the number of samples with the label at 1 in the dataset: it gives n_1 and we note N as the total number of samples in the dataset. Secondly, for each rule $r \in \mathcal{R}_{opt,+}$, we count p_1, the number of samples such that the class is 1, and such that the input fulfilled the rule. We do the same for p_0 and the class 0. Then, we compute the probability:

$$\mathbb{P}(\texttt{C=1|\ r is True}) = \frac{\mathbb{P}(\texttt{C=1})\mathbb{P}(\texttt{r is True|C=1})}{\mathbb{P}(\texttt{r is True|C=1})\mathbb{P}(\texttt{C=1}) + \mathbb{P}(\texttt{r is True|C=0})\mathbb{P}(\texttt{C=0})}$$

Table 1: RDT probabilities associated to the final rules in Figure 2, the Adult dataset case study. $\texttt{Class} = 1$ for ***Rule 1*** and ***Rule 2*** and $\texttt{Class} = 0$ otherwise. $\mathbb{P}(\texttt{Class}|\ \texttt{r is True})$ is contracted to $\mathbb{P}(\texttt{C}|\texttt{r})$.

	$\mathbb{P}(\texttt{C}\|\texttt{r})$	$\mathbb{P}(\texttt{r})$	$\mathbb{P}(\texttt{C}\|\texttt{r, Male})$	$\mathbb{P}(\texttt{r, Male})$	$\mathbb{P}(\texttt{C}\|\texttt{r, Female})$	$\mathbb{P}(\texttt{r, Female})$
Rule 1	0.489	0.297	0.575	0.214	0.266	0.083
Rule 2	0.435	0.456	0.435	0.406	0.441	0.049
Rule 3	0.959	0.022	0.950	0.019	1	0.005
Rule 4	0.833	0.855	0.759	0.569	0.759	0.370
Rule 5	0.920	0.394	0.892	0.257	0.972	0.138

with $\mathbb{P}(\texttt{C=1}) = 1 - \mathbb{P}(\texttt{C=0}) = n_1/N$, $\mathbb{P}(\texttt{r is True}|\texttt{C} = 1) = p_1/n_1$, $\mathbb{P}(\texttt{r is True}|\texttt{C} = 0) = p_0/(n - n_1)$. We also compute $\mathbb{P}(\texttt{C=1}|\ \texttt{r is True and A})$ with $\texttt{A}$ the sensitive attribute. An example is given in Table 1.

4 Experimental results

In this section, we validate quantitatively the above detailed TT-rules model enhancements. We successively present results on four datasets (Section 4.2), a focus on fairness with a benchmark comparison on two datasets (Section 4.3), and a full exploration of one TT-rules $\mathcal{R}_{opt}$ rules on Adult dataset (Section 4.4).

4.1 Experimental set-up

Evaluation measures and training conditions. We use accuracy and statistical parity (defined in Appendix E) as scores to evaluate our model's performances. We also compute the number of rules and define the complexity as the total number of conditions for all rules. A condition refers to an elementary OR/AND condition on individual variables in a rule. Finally, Avg cond./rule is the average number of conditions per rule. All the TTnet training features are detailed in Appendix F. We seeded the experiments and they can be found online on https://github.com/anonymousiclr959/TTrules-main.

Datasets and sensitive attributes. We used two healthcare datasets: Cancer[1] and Diabetes 130 US-Hospitals[2] datasets from the UCI Machine Learning Repository Dua & Graff (2017). We also used two fairness datasets: Adult[3] from the UCI Machine Learning Repository Dua & Graff (2017) and Compas [4] introduced by the ProPublica Angwin et al. (2016). We use Sex for Adult and Race for Compas as the sensitive attributes. Details on datasets are given in Appendix F.3.

4.2 Comparison between most accurate model and smallest model

In Table 2 we are comparing all datasets in the most accurate model and the least complex model between all the configuration of Table 3 in Appendix B. Going from most accurate to simplest, the complexity metric shows substantially lower values with a 38, 2, 31, and 204 factor decrease for the Adult, Compas, Cancer, and Diabetes datasets respectively. The % accuracy is slightly lower $-1.4\%, 0\%, -1.4\%, -0.7\%$ for Adult, Compas, Cancer, and Diabetes respectively.

The results of the ablation study experiments looking at the TT-rules performances on Adult, Compas, Cancer, and Diabetes are presented in Table 3 in Appendix B. We examine the influence of the permutation strategy and rules optimization on our model's behavior.

4.3 Focus on Fairness

Using the Adult and Compas datasets, we have compared our TT-rules model to a series of rule-based models CG Dash et al. (2018), LR Wei et al. (2019), RIPPER Cohen & Singer (1999), and Corels Angelino et al. (2017) as well as two rule-based models optimized for fairness C-LR Zafar et al.

[1] https://archive.ics.uci.edu/ml/datasets/breast+cancer+wisconsin+(diagnostic)

[2] https://archive.ics.uci.edu/ml/datasets/diabetes+130-us+hospitals+for+years+1999-2008

[3] https://archive.ics.uci.edu/ml/datasets/Adult

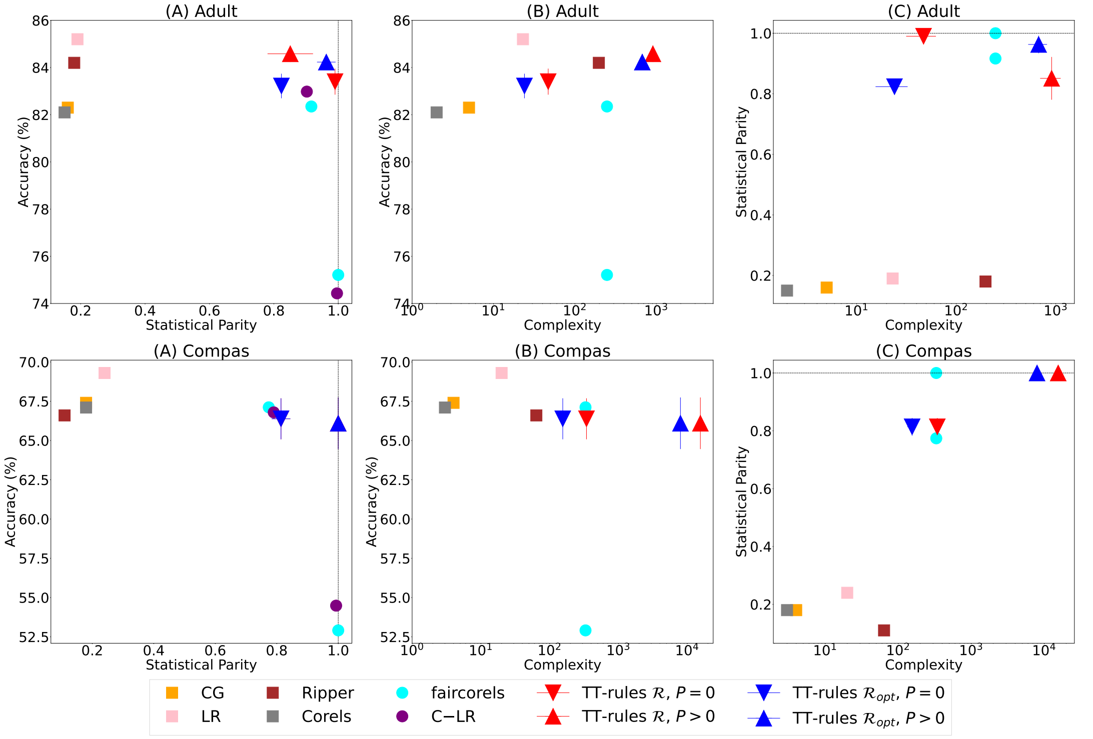
[4] https://github.com/propublica/compas-analysis/raw/master/compas-scores-two-years.csv

Table 2: Comparison between most accurate model and smallest model in TT-rules ablation study. Smallest model refers to the least complex model. All results are computed for 5 models with 5 different k-fold for learning. Δ refers to the difference between the Best and the Smallest model for all metrics.

Dataset	Adult			Compas			Cancer			Diabetes		
Model	Best	Smallest	Δ	Best	Smallest	Δ	Best	Smallest	Δ	Best	Smallest	Δ
Accuracy (%)	84.6	83.2	1.4	66.4	66.4	0.0	97.1	95.7	1.4	57.4	56.7	0.7
Nbr. rules	137	6	131	13	13	0	371	49	322	1059	33	1026
Complexity	909	24	885	343	155	188	6401	205	6196	21644	106	21538
Avg. cond./rule	6	3	3	27	10	16	17	4	12	20	3	17

(2019) and FairCorels Aïvodji et al. (2019). Results are presented Figure 3 as graphics. Baselines training details are presented in Appendix G.

Figure 3: Graphical comparison of our TT-rules model to a series of rule-based models using the Adults or Compas fairness datasets. Triangles points with standard deviations represent our model with red $\mathcal{R}$, blue $\mathcal{R}_{opt}$. Square points are rule-based models (CG, LR, RIPPER, and Corels) and circle points are rule-based models optimized for fairness (C-LR and FairCorels). The first row are graph series based on the Adult dataset and the second row represents the Compas dataset. We charted Accuracy (%) vs Statistical parity, (Graph (A), top right positions for best performances), Accuracy (%) vs Complexity (Graph (B), top left positions for best performances), Statistical parity vs Complexity (Graph (C), and top left positions for best performances). All results are computed for 5 models with 5 different k-fold for learning.



Comparison with state-of-the-art. Rule-set-based models CG Dash et al. (2018), RIPPER Cohen & Singer (1999), and Corels Angelino et al. (2017) have approximately the same complexity as TT-rules and the same performances for both datasets. Only LR Wei et al. (2019), which is not a rule-set-based model, shows better performance with less complexity than our work. On the other hand, our model offers stronger group fairness than all these rule-based models as measured by statistical parity. When comparing the rule-based models optimized for fairness Faircorels Aïvodji et al. (2019) and C-LR Zafar et al. (2019), we observed a fairer behavior than ours, though they are less accurate and more complex on both datasets. All in all, our method based on CNN is equivalent to other rule-based methods while offering the advantageous capability to scale to harder tasks like image classification.

4.4 Case study: Adult datatset

Figure 2 presents the rules in $\mathcal{R}_{opt}$ in ROBBD graph format for one model on Adult dataset and Table 1 presents the corresponding probabilities. We demonstrate the possibility to incorporate post-processing human knowledge through the ability of changing existing rules. We compare our model with Random Forest and decision tree. Complementary results are given in Appendix C.

General: inspecting the rules. We reach 83.6% accuracy with only 4 rules, a complexity of 8 and a statistical parity at 0.824. There are two positive rules which encode for `Class=1` and two negative rules. There is a strong feature selection: rules used only 8 features among the original 100. With the associated probabilities in Table 1, we can distinguish three "types" of rules. First, r_1, r_2 are rules that are activated approximately one third to half of the time ($\mathbb{P}(r_1 \texttt{ is True}) = 0.30$ and $\mathbb{P}(r_2 \texttt{ is True}) = 0.46$) and have a mild impact on the class ($\mathbb{P}(\texttt{Class=1}|r_1 \texttt{ is True}) = 0.49$ and $\mathbb{P}(\texttt{Class=1}|r_2 \texttt{ is True}) = 0.44$). Then, the rule r_3 is activated very rarely ($\mathbb{P}(r_3 \texttt{ is True}) = 0.02$) but is strongly discriminating ($\mathbb{P}(\texttt{Class=0}|r_3 \texttt{ is True}) = 0.96$). Finally, the rule r_4 is often fulfilled and has a strong effect ($\mathbb{P}(r_4 \texttt{ is True}) = 0.83$, $\mathbb{P}(\texttt{Class=0}|r_4 \texttt{ is True}) = 0.86$). Incidentally, this rules-clustering is a starting point for rule ranking, which is the object of future work. With the human improvements of the rules, the accuracy went further up to $84.6\%(+1\%)$, and the statistical parity to $0.838(+0.014)$.

Discussion on individual and group fairness with "Gender" as sensitive feature. This model could appear at first glance as individual gender-fair: in fact, the feature "Gender" does not appear in the decision-making process in any rules. The same stand for highly correlated features such as the "Wife" attribute of the feature "Mariage-status". On the one hand, regarding individual fairness on the sensitive feature "Gender", the model appears fair. On the other hand, regarding group-fairness, the statistical parity score of 0.838 is not maximal. Yet, the probabilities in Table 1 give more information on the reasons why. In fact, rule r_1 advantages Men over Women ($\mathbb{P}(\texttt{Class=1}|r_1\texttt{is True,Male}) = 0.58 > \mathbb{P}(\texttt{Class=1}|r_1\texttt{is True,Female}) = 0.27$. And the negative rules are equally discriminating towards Men and Women (for example, $\mathbb{P}(\texttt{Class=0}|r_4\texttt{is True,Male}) = 0.76 = \mathbb{P}(\texttt{Class=1}|r_4\texttt{is True,Female})$). Because rule r_1 does not appear to be unrealistic nowadays regarding gender, the latter points may underline a possible bias in the dataset.

Discussion on individual fairness with "Race" as sensitive feature: ghost proxies rules. In contrary to the previous paragraph, it is very clear that the model is not individually fair for "Race" sensitive feature. In fact, the first rule that stands out in our model is the BornInMexico $\vee$ BornInNicaragua rule. In a black box model, this rule would have gone unnoticed. Especially if the sensitive feature is "Race", a feature highly correlated to the place of birth. We named them ghost proxies rules as they are usually invisible in a complex model and these rules contain proxies: features that are proxies of the sensitive attribute. These ghost proxy rules, which are highly correlated to a sensitive feature are difficult to find and extremely important to target. In this context, the global and exact interpretability provided by our framework is essential to manage those rules, as well as the little number of rules and the little number of conditions per rule. The detection of "ghost proxy rules" has an application on model portability from one country to another for example: in this case, we clearly understand that this model will need to be changed if we use it in Mexico.

5 Limitations & Conclusion

Limitations. Our method is mostly focused on post-processing, and the only pre-processing amelioration is based on randomness. Moreover, there is no possibility to force some features in some rules, explaining the features in the rules, or ranking the rules. We let these issues for future work.

Conclusion. In this work, we proposed an optimized new CNN-based framework for global and exact interpretability with application to healthcare and fairness tabular datasets. We think that TT-rules is a relevant tool for CNN explainability.

References

Ulrich Aïvodji, Julien Ferry, Sébastien Gambs, Marie-José Huguet, and Mohamed Siala. Learning fair rule lists. *arXiv preprint arXiv:1909.03977*, 2019.

Sheldon B. Akers. Binary decision diagrams. *IEEE Transactions on computers*, 27(06):509–516, 1978.

Elaine Angelino, Nicholas Larus-Stone, Daniel Alabi, Margo Seltzer, and Cynthia Rudin. Learning certifiably optimal rule lists for categorical data. *arXiv preprint arXiv:1704.01701*, 2017.

Julia Angwin, Jeff Larson, Surya Mattu, and Lauren Kirchner. Machine bias. In *Ethics of Data and Analytics*, pp. 254–264. Auerbach Publications, 2016.

Seyed Amir Hossein Aqajari, Emad Kasaeyan Naeini, Milad Asgari Mehrabadi, Sina Labbaf, Nikil Dutt, and Amir M Rahmani. pyeda: An open-source python toolkit for pre-processing and feature extraction of electrodermal activity. *Procedia Computer Science*, 184:99–106, 2021.

Vijay Arya, Rachel K. E. Bellamy, Pin-Yu Chen, Amit Dhurandhar, Michael Hind, Samuel C. Hoffman, Stephanie Houde, Q. Vera Liao, Ronny Luss, Aleksandra Mojsilović, Sami Mourad, Pablo Pedemonte, Ramya Raghavendra, John Richards, Prasanna Sattigeri, Karthikeyan Shanmugam, Moninder Singh, Kush R. Varshney, Dennis Wei, and Yunfeng Zhang. One explanation does not fit all: A toolkit and taxonomy of ai explainability techniques, 2019. URL https://arxiv.org/abs/1909.03012.

Mislav Balunovic, Anian Ruoss, and Martin Vechev. Fair normalizing flows. In *International Conference on Learning Representations*, 2021.

Adrien Benamira, Thomas Peyrin, and Bryan Hooi Kuen-Yew. Truth-table net: A new convolutional architecture encodable by design into sat formulas. *arXiv preprint arXiv:2208.08609*, 2022.

Christian Bessiere, Emmanuel Hebrard, and Barry O'Sullivan. Minimising decision tree size as combinatorial optimisation. In *International Conference on Principles and Practice of Constraint Programming*, pp. 173–187. Springer, 2009.

Armin Biere, Marijn Heule, and Hans van Maaren. *Handbook of satisfiability*, volume 185. IOS press, 2009.

Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

Randal E Bryant. Graph-based algorithms for boolean function manipulation. *Computers, IEEE Transactions on*, 100(8):677–691, 1986.

William W Cohen. Fast effective rule induction. In *Machine learning proceedings 1995*, pp. 115–123. Elsevier, 1995.

William W Cohen and Yoram Singer. A simple, fast, and effective rule learner. *AAAI/IAAI*, 99 (335-342):3, 1999.

European Commission. Proposal for a regulation laying down harmonised rules on artificial intelligence. 2021. URL https://digital-strategy.ec.europa.eu/en/library/proposal-regulation-laying-down-harmonised-rules-artificial-intelligence.

Sanjeeb Dash, Oktay Gunluk, and Dennis Wei. Boolean decision rules via column generation. *Advances in neural information processing systems*, 31, 2018.

Michael Driscoll. System and method for adapting a neural network model on a hardware platform, July 2 2020. US Patent App. 16/728,884.

Mengnan Du, Subhabrata Mukherjee, Guanchu Wang, Ruixiang Tang, Ahmed Awadallah, and Xia Hu. Fairness via representation neutralization. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 12091–12103. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper/2021/file/64ff7983a47d331b13a81156e2f4d29d-Paper.pdf.

Dheeru Dua and Casey Graff. UCI machine learning repository, 2017. URL `http://archive.ics.uci.edu/ml`.

Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. Fairness through awareness. In *Proceedings of the 3rd innovations in theoretical computer science conference*, pp. 214–226, 2012.

Alex A Freitas. Comprehensible classification models: a position paper. *ACM SIGKDD explorations newsletter*, 15(1):1–10, 2014.

Nir Friedman, Dan Geiger, and Moises Goldszmidt. Bayesian network classifiers. *Machine learning*, 29(2):131–163, 1997.

Daniel Fryer, Inga Strümke, and Hien Nguyen. Shapley values for feature selection: the good, the bad, and the axioms. *IEEE Access*, 9:144352–144360, 2021.

Moritz Hardt, Eric Price, and Nati Srebro. Equality of opportunity in supervised learning. *Advances in neural information processing systems*, 29, 2016.

Congjie He, Meng Ma, and Ping Wang. Extract interpretability-accuracy balanced rules from artificial neural networks: A review. *Neurocomputing*, 387:346–358, 2020.

Kai Jia and Martin Rinard. Efficient exact verification of binarized neural networks. *arXiv preprint arXiv:2005.03597*, 2020.

Faisal Kamiran and Toon Calders. Data preprocessing techniques for classification without discrimination. *Knowledge and information systems*, 33(1):1–33, 2012.

Toshihiro Kamishima, Shotaro Akaho, Hideki Asoh, and Jun Sakuma. Fairness-aware classifier with prejudice remover regularizer. In Peter A. Flach, Tijl De Bie, and Nello Cristianini (eds.), *Machine Learning and Knowledge Discovery in Databases*, pp. 35–50, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg. ISBN 978-3-642-33486-3.

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

Matt J Kusner, Joshua Loftus, Chris Russell, and Ricardo Silva. Counterfactual fairness. *Advances in neural information processing systems*, 30, 2017.

Himabindu Lakkaraju, Stephen H Bach, and Jure Leskovec. 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.

Chang-Yeong Lee. Representation of switching circuits by binary-decision programs. *The Bell System Technical Journal*, 38(4):985–999, 1959.

Daniel McNamara, Cheng Soon Ong, and Robert C Williamson. Costs and benefits of fair representation learning. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, pp. 263–270, 2019.

Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. A survey on bias and fairness in machine learning. *ACM Computing Surveys (CSUR)*, 54(6):1–35, 2021.

Christoph Molnar. *Interpretable machine learning*. Lulu. com, 2020.

Osonde A Osoba and William Welser IV. *An intelligence in our image: The risks of bias and errors in artificial intelligence*. Rand Corporation, 2017.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32: 8026–8037, 2019.

Geoff Pleiss, Manish Raghavan, Felix Wu, Jon Kleinberg, and Kilian Q Weinberger. On fairness and calibration. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL `https://proceedings.neurips.cc/paper/2017/file/b8b9c74ac526fffbeb2d39ab038d1cd7-Paper.pdf`.

Willard V Quine. The problem of simplifying truth functions. *The American mathematical monthly*, 59(8):521–531, 1952.

J. Ross Quinlan. Induction of decision trees. *Machine learning*, 1(1):81–106, 1986.

J. Ross Quinlan. Simplifying decision trees. *International journal of man-machine studies*, 27(3): 221–234, 1987.

J Ross Quinlan. *C4. 5: programs for machine learning*. Elsevier, 2014.

General Data Protection Regulation. Regulation eu 2016/679 of the european parliament and of the council of 27 april 2016. *Official Journal of the European Union. Available at: http://ec. europa. eu/justice/data-protection/reform/files/regulation_oj_en. pdf (accessed 20 September 2017)*, 2016.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. " 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*, pp. 1135–1144, 2016.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Anchors: High-precision model-agnostic explanations. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Ronald L Rivest. Learning decision lists. *Machine learning*, 2(3):229–246, 1987.

Tatiana Tommasi, Novi Patricia, Barbara Caputo, and Tinne Tuytelaars. A deeper look at dataset bias. In *Domain adaptation in computer vision applications*, pp. 37–55. Springer, 2017.

Antonio Torralba and Alexei A Efros. Unbiased look at dataset bias. In *CVPR 2011*, pp. 1521–1528. IEEE, 2011.

Zhuo Wang, Wei Zhang, Ning Liu, and Jianyong Wang. Scalable rule-based representation learning for interpretable classification. *Advances in Neural Information Processing Systems*, 34, 2021.

Dennis Wei, Sanjeeb Dash, Tian Gao, and Oktay Gunluk. Generalized linear rule models. In *International Conference on Machine Learning*, pp. 6687–6696. PMLR, 2019.

Fan Yang, Kai He, Linxiao Yang, Hongxia Du, Jingbang Yang, Bo Yang, and Liang Sun. Learning interpretable decision rule sets: A submodular optimization approach. *Advances in Neural Information Processing Systems*, 34, 2021.

Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez-Rodriguez, and Krishna P Gummadi. Fairness constraints: A flexible approach for fair classification. *The Journal of Machine Learning Research*, 20(1):2737–2778, 2019.

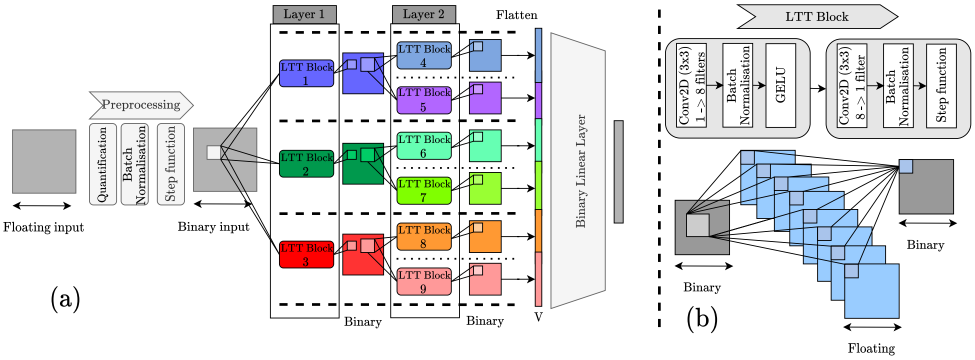
Yu Zhang, Peter Tiňo, Aleš Leonardis, and Ke Tang. A survey on neural network interpretability. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2021.

A TTnet General 2D-CNN Architecture

We present in Figure 4 the general architecture of a 2D-CNN TTnet.

The TTnet architecture Benamira et al. (2022) is composed of LTT blocks. One LTT block can be seen as one filter of a CNN and has its own SAT expression. A LTT block is composed of two layers of grouped 2D-CNN with an expanding factor of 8. It can be envisioned as an expanding auto-encoder, with only 1 channel input. The intermediate values are real and the input/output values are binary. Finally, the preprocessing TTnet block is composed of a batch normalization layer and a step function. The TTnet classification layer is a binary linear layer. We refer to the original article for more details.

Figure 4: **(a)** General architecture of the TTnet model with a one-channel input. Layer 0 is a pre-processing layer that allows image binarization. Then follow two layers of Learning Truth Table (LTT) blocks: three blocks in the first layer, six in the second. It should be noted that the LTT block of layer 2 does not take as input all the filters of layer 1, as it is usually the case: it only takes the filter of their groups. Finally, the last linear layer performs the classification. **(b)** Architecture of Learning Truth Table (LTT) block. A LTT block is composed of two layers of grouped 2D-CNN with an expanding factor of 8. It can be seen as an expanding auto-encoder. The intermediate values are real and the input/output values are binary.



B Complementary Results Ablation study

We present additional results for the ablation study in Table 3.

Permutation strategy influence. Except for Compas, the algorithm is valuable for $P \geq 5$. Indeed, the accuracy increases by $1.2\%, 0.7\%, 0.4\%$ for the datasets Adult, Cancer, and Diabetes respectively when using the permutation strategy. On the other hand, it drastically amplifies the complexity, with a 10 to 40-fold increase, depending on the dataset. We also note that the permutation strategy has barely any impact on the average number of conditions per rule. This is expected as this metric is heavily depending on the kernel size of TTnet.

Rules optimizations influence. Complexity decreases drastically from $\mathcal{R}$ to $\mathcal{R}_{opt}$: between $\times 1.4$ and $\times 2.8$ for Adult, between $\times 1.9$ and $\times 2.3$ for Compas, between $\times 1.8$ and $\times 5.3$ for Cancer and between $\times 1.4$ and $\times 5.4$ for Diabetes. On the other hand, except for Compas, the optimizations lead to a slight decrease in measured accuracy, the largest observed decreases being -0.5%, -0.7%, and -0.9% for the datasets Adult, Cancer, and Diabetes respectively. In addition, we can observe that the size of each rule decreases: this is mainly due to the DCT injection. Quine-McCluskey algorithm Quine (1952) outputs smaller rules with DCT usage. Finally, in the case $P = 10$, many rules are filtered, even with the high threshold ± 0.9. These optimizations are only possible for TT-rules.

C Complementary results on Adult Dataset use-case

Changing existing rules & adding a new rule. We present in Table 5 the results of all the modifications depicted in Figure 2. We first change the existing rules by, on the one hand, optimizing numerical values of existing conditions and on the other hand adding complementary conditions. We optimized on the training dataset the numerical values in the "Years of Education" condition, and we fixed it to 12. Then, we added the conditions "Is a School Prof" in Rule 2. In fact, it seems to be an important feature as we find it in almost all other TT-rules configurations. We also added "Born In Peru" in Rule 3 as we simply add the countries of South and Latin America (other such countries did not change performances so we did not include them). The accuracy went up to 83.66% and 83.61% respectively and the statistical parity to 0.823 and 0.824 respectively. These changes are done on the train, altogether leading to test the accuracy of 84.0% $(+0.4\%)$ and a statistical parity of 0.826 $(+0.02)$. Surprisingly, age and job type do not influence the prediction. We also noticed that the "Statistical Weight" feature is used in most of the models. Therefore, we decided to add the following rule displayed in Figure 2 : $(\text{Age} < 28) \vee \text{Farmer} \vee \text{HandlerCleaner} \vee \text{Machinist} \vee ((\text{Age} >$

Table 3: Ablation study results for TT-rules on the four datasets. All results are computed for 5 models with 5 different k-fold for learning. No permutation column refers to normal dataset. Complexity is defined as the total number of conditions for all rules.

Dataset	**Permutations**	**No Permutation**		$P = 5$		$P = 10$	
	Rule Set used	$\mathcal{R}$	$\mathcal{R}_{opt}$	$\mathcal{R}$	$\mathcal{R}_{opt}$	$\mathcal{R}$	$\mathcal{R}_{opt}$
Adult	Accuracy (%)	83.4 ± 0.6	83.2 ± 0.5	84.6 ± 0.3	84.2 ± 0.3	84.4 ± 0.5	83.9 ± 0.3
	F1-score (%)	61.1 ± 1.3	61.1 ± 1.4	65.9 ± 1.4	66.1 ± 1.6	66.9 ± 0.6	64.0 ± 5.7
	Number of rules	7.0 ± 2.2	6.0 ± 2.2	137.0 ± 6.8	130.0 ± 9.8	310.0 ± 27.8	260.0 ± 23.0
	Complexity	47.0 ± 15.6	24.0 ± 8.5	909.0 ± 212.2	673.0 ± 144.5	2156.0 ± 374.5	1404.0 ± 181.7
	Avgerage conditions/rule	5.6 ± 1.2	3.3 ± 1.5	6.7 ± 1.4	5.1 ± 0.9	7.8 ± 0.9	5.8 ± 0.4
	Statistical Parity	99.0 ± 0.6	82.4 ± 1.1	85.1 ± 7.1	96.3 ± 2.9	93.8 ± 2.5	nan ± nan
	Odds Equalized	0.0 ± 0.0	0.0 ± 0.0	-0.1 ± 0.0	-0.1 ± 0.1	-0.1 ± 0.0	-0.1 ± 0.1
	Extraction time (s.)	7 ± 2	8 ± 1	9 ± 1	10 ± 2	17 ± 2	17 ± 4
	Train Time (s.)	53 ± 2		56 ± 3		58 ± 1	
Compas	Accuracy (%)	66.4 ± 1.3	66.4 ± 1.3	65.5 ± 1.2	65.5 ± 1.2	66.1 ± 1.6	66.1 ± 1.6
	F1-score (%)	72.2 ± 1.7	72.2 ± 1.7	71.3 ± 0.7	71.3 ± 0.7	70.8 ± 2.1	70.8 ± 2.1
	Number of rules	13.0 ± 1.8	13.0 ± 1.8	219.0 ± 9.9	219.0 ± 9.9	511.0 ± 9.5	511.0 ± 9.5
	Complexity	343.0 ± 41.0	155.0 ± 21.8	6118.0 ± 812.3	2670.0 ± 468.4	15541.0 ± 1276.7	7974.0 ± 918.0
	Avgerage conditions/rule	27.1 ± 3.2	10.3 ± 1.3	27.9 ± 2.4	12.2 ± 1.6	30.4 ± 2.4	15.6 ± 1.8
	Statistical Parity	81.4 ± 3.0	81.4 ± 3.0	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0
	Odds Equalized	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.1	0.0 ± 0.1	0.1 ± 0.1	0.1 ± 0.1
	Extraction time (s.)	7 ± 1	7 ± 1	12 ± 1	12 ± 1	22 ± 1	21 ± 1
	Train Time	117 ± 7		118 ± 3		117 ± 8	
Cancer	Accuracy (%)	96.4 ± 1.0	95.7 ± 0.9	95.7 ± 0.9	95.7 ± 0.7	97.1 ± 0.6	97.1 ± 0.7
	F1-score (%)	94.9 ± 1.6	94.3 ± 1.6	94.6 ± 1.4	93.8 ± 1.0	96.0 ± 1.0	96.0 ± 1.2
	Number of rules	60.0 ± 3.6	49.0 ± 6.8	288.0 ± 8.7	231.0 ± 21.5	615.0 ± 22.0	371.0 ± 61.8
	Complexity	1098.0 ± 239.5	205.0 ± 33.7	5222.0 ± 167.4	2924.0 ± 328.5	10467.0 ± 1902.1	4665.0 ± 1027.4
	Avgerage conditions/rule	19.4 ± 2.8	4.2 ± 0.3	17.6 ± 0.6	13.3 ± 0.8	17.2 ± 2.5	12.1 ± 1.1
	Extraction time (s.)	9 ± 1	9 ± 1	18 ± 1	18 ± 1	31 ± 5	30 ± 5
	Train Time (s.)	19 ± 1		20 ± 2		20 ± 2	
Diabetes	Accuracy (%)	56.7 ± 0.2	56.7 ± 0.2	57.0 ± 0.5	57.0 ± 0.5	57.4 ± 0.8	56.5 ± 0.9
	F1-score (%)	56.7 ± 0.2	56.7 ± 0.2	57.0 ± 0.5	57.0 ± 0.5	57.4 ± 0.8	56.5 ± 0.9
	Number of rules	35.0 ± 5.9	33.0 ± 5.2	406.0 ± 15.7	406.0 ± 15.7	1059.0 ± 45.5	1043.0 ± 46.2
	Complexity	568.0 ± 131.9	106.0 ± 27.6	7834.0 ± 1143.0	5561.0 ± 761.1	21644.0 ± 2802.2	15475.0 ± 2224.5
	Avgerage conditions/rule	16.2 ± 1.3	3.6 ± 0.4	19.4 ± 2.5	13.8 ± 1.7	20.8 ± 2.8	14.3 ± 1.7
	Extraction time (s.)	19 ± 3	18 ± 2	35 ± 6	34 ± 5	68 ± 4	67 ± 5
	Train Time (s.)	110 ± 4		126 ± 8		149 ± 4	

$60) \wedge (\text{StatisticalWeight} < 300k))$. The accuracy went further up to $84.6\%(+1\%)$, and the statistical parity to $0.838 + (+0.01)$. This illustrates the possibility to add human knowledge to a CNN-based model.

Metrics affected by rule changes. In Table 5, we present the metrics affected by the different rule changes: the accuracy and statistical parity increases with each changes. The number of conditions increases also because of the small added rule, but remains fairly low.

Effect of the new rule on Statistical Parity. When looking at the probabilities in Table 1, we may not understand why the Statistical Parity increases when Rule 5 is added. Therefore, we computed the Statistical Parity for the following scenarios: once without Rule 3, once without Rule 4, and once without Rule 5. The results are available in Table 4. We can observe that all of those rules have a positive impact on this metric, as the metric decreases when one of the rules is removed. Moreover, with $\Delta = |\mathbb{P}(\texttt{C}|\texttt{r},\texttt{Male}) * \mathbb{P}(\texttt{r},\texttt{Male}) - \mathbb{P}(\texttt{C}|\texttt{r},\texttt{Female}) * \mathbb{P}(\texttt{r},\texttt{Female})|$ increasing, the Statistical Parity increase. If Δ is small, then the events $(\texttt{r},\texttt{Male})$ and $(\texttt{r},\texttt{Female})$ are likely to describe similar subgroups. If Δ is high, then the two subgroups are likely to differ.

Table 4: Statistical Parity of the model presented in Figure 2 without the rules 3, 4 or 5. As a remainder, the Statistical Parity of the model with all those rules is 83.83%. We denote $\Delta = |\mathbb{P}(\texttt{C}|\texttt{r},\texttt{Male}) * \mathbb{P}(\texttt{r},\texttt{Male}) - \mathbb{P}(\texttt{C}|\texttt{r},\texttt{Female}) * \mathbb{P}(\texttt{r},\texttt{Female})|$. The smaller is Δ, the smaller is the impact of the Rule on the Statistical Parity.

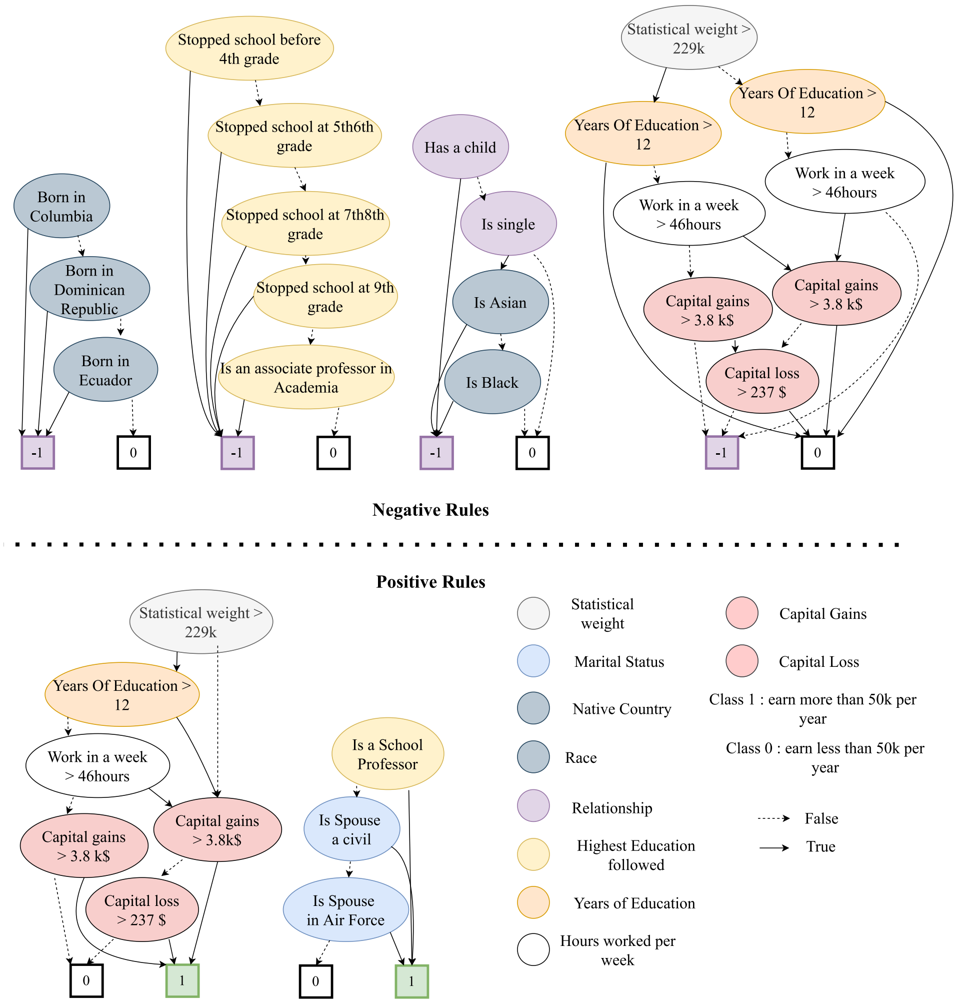
Model Without	Rule 3	Rule 4	Rule 5
Statistical Parity %	83.69	64.81	82.49
Δ	0.0140	0.1511	0.093

Other Model for Adult. Regarding this specific model in Figure 2, the rules are fully interpretable. Yet, depending on the features learnt by TTnet, it would be sometimes not fully interpretable: indeed, the feature Statistical Weight is a feature regarding the sampling weight of each group in the Adult census, see Figure 5. This feature does not give a clear indication about an individual. Moreover, some clear unfair rules regarding the Race attribute were deducted by the algorithm such as the

Table 5: Metrics of the initial model in Figure 2 for all different rules changes, the added rule, and those changes altogether. All those changes improve the accuracy and the statistical parity except for the Rule 3 which only increases the accuracy.

	No Changes	Rule 2 Changed	Rule 3 Changed	Rule 4 Changed	One Rule Added	All Changes
Accuracy %	83.60	84.01	83.66	83.61	84.17	84.61
Complexity	8	8	9	9	13	15
Stat. parity %	82.35	82.68	82.30	82.68	83.85	83.83

Figure 5: An other example of a learned model on the Adult dataset. With this model, we reach an accuracy of 82.28% but with much more conditions than the initial one presented in Figure 2 which reached an accuracy of 83.6%. They have 29 conditions and 8 conditions respectively.



Born in Columbia/Dominican Republic/Ecuador features. These features are not directly the Race attributes, but are correlated to the Race of the individual. Here, it would translate to a ghost rule describing a Latin American. This model even learnt the Black and Asian attribute with an accuracy of 82.28%. By removing these two conditions, we got an accuracy of 82.43%. These rules are clearly affected by inter-attributes correlation. As such, the original dataset and the features used have to be carefully analysed to ensure a fair algorithm.

D RULES DISTRIBUTION TABLE ALGORITHM

We give in Algorithm 1 the process used to construct the Rules Distribution Table from a dataset, .

Algorithm 1 Build RDT function used to construct the Rules Distribution Table (RDT) from a dataset $\mathcal{D}$ and a set of relevant rules $\mathcal{R}_+$.

```
function BUILD RDT($\mathcal{D}, \mathcal{R}_+$)
    RDT = {}
    $n = |\mathcal{D}|$
    $n_1 = \texttt{card( (I,L)} \in \mathcal{D}| \texttt{ L } = 1)$        ▷ Number of samples with label 1 in the dataset.
    $P(\texttt{Class=1}) = P(\texttt{C=1}) = n_1/n$
    $P(\texttt{Class=0}) = P(\texttt{C=0}) = 1 - P(\texttt{C=1})$
    for all $r$ in $\mathcal{R}_+$ do
        $p_1, p_0 = 0, 0$
        for all $(I, L)$ in $D$ do
            if $L = 1$ and $r(I)$ is True then        ▷ $r(I)$ is True: the input $I$ satisfies the rule $r$
                $p_1 + = 1$
            end if
            if $L = 0$ and $r(I)$ is True then
                $p_0 + = 1$
            end if
        end for
        $P(\texttt{r is True}|C = 1) = p_1/n_1$
        $P(\texttt{r is True}|C = 0) = p_0/(n - n_1)$
        $P(\texttt{r is True}) = P(\texttt{r is True}|\texttt{C=1})P(\texttt{C=1}) + P(\texttt{r is True}|\texttt{C=0})P(\texttt{C=0})$
        $P(\texttt{C=1}|\texttt{r is True}) = \frac{P(\texttt{C=1})P(\texttt{r is True}|\texttt{C=1})}{P(\texttt{r is True}|\texttt{C=1})P(\texttt{C=1}) + P(\texttt{r is True}|\texttt{C=0})P(\texttt{C=0})}$
        RDT[$r$] = $[P(\texttt{C=1}|\texttt{r is True}), P(\texttt{r is True})]$
    end for
    return RDT
end function
```

E FAIRNESS CRITERIA

We consider a dataset sample $X \in \mathbb{R}^d$ with $d \in \mathbb{N}$ as the sample size, with a sensitive attribute $a \in \{0, 1\}$, labels $Y \in \{0, 1\}$ and a classifier $h : \mathbb{R}^d \mapsto \{0, 1\}, X \mapsto h(X) = \hat{Y}$. We denote $\hat{Y}_{a \leftarrow i} = h(X_{a \leftarrow i})$ the prediction of sample X, with the sensitive attribute forced to $i \in \{0, 1\}$. One fairness metric is considered here: the Statistical Parity metric for quantifying group fairness. The Statistical Parity metric is defined in Dwork et al. (2012) as $1 - |\mathbb{P}(\hat{Y} = 1|a = 0) - \mathbb{P}(\hat{Y} = 1|a = 1)|$. The closer to 1 the metric is, the more likely the classifier will have the same performance on the two sensitive groups of people. Maximizing group fairness metrics does not guarantee that, for a given model, each individual has the same model prediction whatever the value of its sensitive feature: this is individual fairness. Individual fairness is hard to quantify as there can always exist features that are proxies of the sensitive attribute Kusner et al. (2017), which is hard to quantify. In this paper, we propose to quantify group-fairness and, as we can compute all the rules, we propose to manually investigate the individual fairness on a concrete use-case model on Adult dataset. Automatic detection of individual unfairness rules is out of the scope of this paper.

F ON REPRODUCTIBILITY

F.1 MODEL ARCHITECTURE AND TRAINING CONDITIONS - GENERAL

Experimental environment. The project was implemented with Python and the library PyTorch Paszke et al. (2019). Our workstation is constituted of a GPU Nvidia GeForce GTX 970 with 4043 MiB memory and four Intel core i5-4460 processors clocked at 3.20 GHz.

Training method. We built our training method on top of Jia & Rinard (2020) project and we refer to their notations for this section. We trained the networks using the Adam optimizer Kingma & Ba (2014) for 10 epochs with a minibatch size of 128. The mean and variance statistics of batch normalization layers are recomputed on the whole training set after training finishes.

Weights initialization. Weights for the final connected layers are initialized from a Gaussian distribution with standard deviation 0.01 and the mask weights in BinMask are enforced to be positive by taking the absolute value during initialization.

Other hyperparameters. We apply a weight decay of $1e-7$ on the binarized mask weight of BinMask.

Architecture. Each dataset has the same input layers and output layer, with of course the size adapted to the dataset for the output layer. The first layer is a Batch Normalization layer with an ϵ of 10^{-5} and a momemtum of 0.1. It is followed by the LTT block, which will be detailled for each dataset. Finally, the binary linear regression is computed, with input and output features sizes detailled below.

F.2 Model Architecture and Training conditions - Dataset specific

Adult. We trained the Adult dataset in 10 epochs. The first layer is the Batch Normalization layer as described above. It is followed by a LTT block. The first convolution has 10 filters. The convolution is done with a stride of 5, a kernel size of 5 and no padding. It is followed by a Batch Normalization layer with the same parameters as the input one. The following convolution has 10 filters and a kernel size and a stride of 1. A last Batch Normalization finishes with the same parameter as above the LTT Block. The final linear regression takes 200 features as input and 2 as outputs. The learning rate is 0.005.

Compas. We trained the Compas dataset in 60 epochs. The first layer is the Batch Normalization layer as described above. It is followed by a LTT block. The first convolution has 20 filters (amplification of 20). The convolution is done with a stride of 1, a kernel size of 6 and no padding. It is followed by a Batch Normalization layer with the same parameters as the input one. The following convolution has 5 filters and a kernel size and a stride of 1. A last Batch Normalization finishes with the same parameter as above the LTT Block. The final linear regression takes 60 features as input and 2 as outputs. The learning rate is 0.0005. Only the best model in terms of testing accuracy was kept.

Cancer. We trained the Cancer dataset in 10 epochs. The first layer is the Batch Normalization layer as described above. It is followed by a LTT block. The first convolution has 10 filters (amplification of 10). The convolution is done with a stride of 1, a kernel size of 6 and no padding. It is followed by a Batch Normalization layer with the same parameters as the input one. The following convolution has 10 filters and a kernel size and a stride of 1. A last Batch Normalization finishes with the same parameter as above the LTT Block. The final linear regression takes 80 features as input and 2 as outputs. The learning rate is 0.005.

Diabetes. We trained the Diabetes dataset in 10 epochs. The first layer is the Batch Normalization layer as described above. It is followed by a LTT block. The first convolution has 10 filters (amplification of 10). The convolution is done with a stride of 1, a kernel size of 6 and no padding. It is followed by a Batch Normalization layer with the same parameters as the input one. The following convolution has 10 filters and a kernel size and a stride of 1. A last Batch Normalization finishes with the same parameter as above the LTT Block. The final linear regression takes 295 features as input and 3 as outputs. The learning rate is 0.0005.

F.3 Dataset details

All datasets have been split 5 times in a 80-20 train-test split for k-fold testing.

Adult. The Adult dataset contains 48,842 individuals with 18 binary features and a label indicating whether the income is greater than 50K$ USD or not.

Compas. The Compas dataset consists of 6,172 individuals with 10 binary features and a label that takes a value of 1 if the individual does not re-offend and 0 otherwise.

Cancer. Cancer dataset contains 569 data points with 30 numerical features. The goal is to predict if a tumor is cancerous or not. We encoded each integer value into a one-hot vector, resulting in a total of 81 binary features.

Diabetes. Regarding the Diabetes dataset, it contains 100000 data points of patients with 50 features, both categorical and numerical. We kept 43 features, 5 numerical and the other are categorical which resulted in 291 binary features and 5 numerical features. We will predict one of the three labels for hospital readmission.

G BASELINES IMPLEMENTATIONS DETAILS

Rule-based models. We used the implementations proposed in Arya et al. (2019). We implement a cross-validation to find the corresponding relevant hyper-parameters. We also evaluate the baselines on the same 5-fold datasets.

Rule-based models optimized for fairness. We used the results proposed in Aïvodji et al. (2019).