A comparative analysis of rule-based, model-agnostic methods for explainable artificial intelligence

Giulia Vilone[1 [0000−0002−4401−5664]], Lucas Rizzo[1 [0000−0001−9805−5306]], and Luca Longo[1 [0000−0002−2718−5426]]

School of Computer Science, Technological University Dublin.
{giulia.vilone, lucas.rizzo, luca.longo}@tudublin.ie

Abstract. The ultimate goal of Explainable Artificial Intelligence is to build models that possess both high accuracy and degree of explainability. Understanding the inferences of such models can be seen as a process that discloses the relationships between their input and output. These relationships can be represented as a set of inference rules which are usually not explicit within a model. Scholars have proposed several methods for extracting rules from data-driven machine-learned models. However, limited work exist on their comparison. This study proposes a novel comparative approach to evaluate and compare the rulesets produced by four post-hoc rule extractors by employing six quantitative metrics. Findings demonstrate that these metrics can actually help identify superior methods over the others thus are capable of successfully modelling distinctively aspects of explainability.

Keywords: Explainable artificial intelligence · Rule extraction · Method comparison and evaluation.

1 Introduction

Explainable Artificial Intelligence (XAI) has emerged as an important sub-field of Artificial Intelligence, aimed at building methods and techniques to learn predictive models from data that possess high accuracy and a high degree of explainability. Explainability can be seen as the degree of transparency and understandability of the functioning of a model as perceived by end-users. The explosion of data availability and the advances of machine learning and deep learning have led to the fast development of new models in a variety of domains. Unfortunately, most of these are considered as 'black-box' with underlying complex structures that are difficult to explain to end-users. As a consequence, a number of approaches have emerged to extract information from trained models and try to reconstruct their inference process [9, 21]. Some of these methods are model agnostic and, theoretically, suitable for building a global layer of explanation. However, research studies have demonstrated that this is a difficult task as they merely extract a set of rules that attempts to achieve the same

inference of the underlying model [9]. Furthermore, these rules are not necessarily consistent with a user's domain knowledge. Rather, they might be based on spurious correlations of an input dataset. Recent studies have tried to solve these issues by integrating machine-learned models and symbolic representation of knowledge. For instance, symbolic rules are extracted from trained models in [5]; however, they are built upon a set of symbols not easily interpretable by non-experts. Other attempts are based upon the generation of if-then rules that should be easily readable and understandable by humans, thus adding a meaningful descriptive layer to the underlying model [17]. Nonetheless, little has been done to assess the degree of explainability of these rules in an objective and quantitative manner. This study aims at filling this gap by evaluating XAI methods for rule extractions and compare their explainability across a number of metrics. The machine-learned models were trained on datasets with handcrafted features. The underlying research question of this research is *"To what extent can a set of quantitative metrics be formed and employed to assess and compare the degree of explainability of if-then rule extraction methods?"*.

The remainder of this manuscript is organised as it follows. Section 2 provides a description of the strategies used by scholars to generate explanations of machine learned models, with a focus on rule-extraction algorithms. Section 3 describes the design of a secondary research experiment and the metrics employed to evaluate and compare the degree of explainability of rulesets extracted by four XAI post-hoc model agnostic methods. Section 4 discusses the findings obtained from the experiment. Eventually, Section 5 emphasises the contribution to knowledge and sets future directions.

2 Related work

Over time, researchers have tried to comprehend and explain the inner mechanics of data-driven machine-learned models in various ways [19, 14]. Thus, several types of XAI methods have been proposed. They can be clustered according to the *scope* of an explanation, the *stage* at which a method generates explanations, the *input data* of the underlying model, and the *output format* of the explanation itself [23]. Methods for explainability with a *global* scope attempt to make the entire inferential process of a model transparent and understandable, whereas *local* methods explain it around a specific input instance. *Ante-hoc* methods tackle the explainability of a model from its implementation and during training. The goal is to make it naturally explainable while still trying to reach optimal accuracy and minimal error. *Post-hoc* methods instead keep a trained model unchanged and mimic or explain its behaviour by using an external explainer at testing time. The format of the input data (numerical/categorical, pictorial, textual or times series) of a model can play an important role in constructing a method for explainability as the logic followed by its inferential process can vary according to the inputs, thus requiring different formats of explanations (numerical, *rules*, textual, visual or mixed). Several rule extraction methods exist in the literature, but this study utilised only four of them selected according to three criteria

listed in Section 3. Rule Extraction From Neural Network Ensemble (REFNE) was originally designed to extract symbolic rules from trained neural network ensembles, but its application can be also extended to other learning approaches [26]. Once the original labels of a training dataset are replaced with those generated by the ensemble, REFNE selects a categorical feature and checks if there is a value such that all the instances possessing it fall into the same class. If this condition is satisfied, a rule is created with the value as antecedent. Otherwise, the algorithm selects another categorical input feature, combines its values with those of the feature previously selected and checks if it is possible to create new rules with two antecedents. Rules are limited to only three antecedents [26]. When all the categorical features have been examined, the continuous ones are discretised with the ChiMerge discretization algorithm and considered as new categorical features. The process terminates when no more rules can be created. An alternative method which extracts if-then rules from neural network ensembles is C4.5Rule-PANE [25]. It uses the C4.5 rule induction algorithm to build a ruleset to mimic the inferential process of an ensemble from a training dataset whose original labels have been replaced with those predicted by the ensemble. Similarly, the third method, TREPAN, induces a decision tree by querying the underlying network to determine the output class of each instance [7, 6]. Subsequently, it splits each node of the tree by using the gain ratio criterion. In addition, it considers as constraints the previously selected splits that lie on the path from the root to that node. Another method is Rule Extraction by Reverse Engineering (RxREN) which relies on a reverse engineering technique [4]. It traces back input features that lead to the final result, whilst pruning the insignificant input neurons. Afterwards, it determines the data ranges of each significant neuron in each output class. This is done by iteratively removing one input input feature at the time and measuring the impact on the number of misclassified instances. This process can be seen as a feature selection approach and it is easily applicable to other architectures. The algorithm is recursive and generates hierarchical if-then rules where conditions for discrete attributes are disjoint from the continuous ones.

Scholars have identified various attributes that might affect the degree of explainability of a ruleset [10–12, 15]. Among them, *attribute costs* represent the computational effort to get access to the actual value of an attribute of the data. For example, it is easy to assess the gender of a patient but some health-related attributes can require an expensive investigation. Rules that utilise only attributes based on easily-accessible data are more appealing as they help in keeping the costs low. The interestingness of a rule must take into account the *misclassification costs*. In some domains of application, the erroneous classification of an instance might have a significant impact, not only in economical terms, but also in terms of human lives. To be measured in a quantitative manner, these factors require the integration of user's domain knowledge which is a manual and time-consuming process. Luckily, scholars have identified several other factors of explainability that can be assessed in an objective manner, meaning that they just need the information provided by the data, without relying

on domain knowledge [2, 1, 20, 16]. For example, the number of rules and the number of antecedents of each rule should be minimised as conciseness is a key factor of interpretability [16]. Beyond these attributes, scholars proposed other requirements, or validation factors, to be met by every type of explanation automatically generated by a XAI method. These include, for instance, the correctness of a ruleset, measured as the portion of the dataset correctly classified by rules, must be maximised to generate trustable explanations [15].

3 Design

The subset of XAI methods generating if-then rules from the inferences of machine-learned models is quite large, so it was necessary to narrow it down by adding the following three inclusion criteria:

1. The methods must be *model-agnostic*, meaning that they do not consider the internal components of a model such as weights or structural information, therefore they can be applied to any black-box model. This limits the choice to the post-hoc methods as the ante-hoc ones are inherently model-specific.
2. The rule-extractors must consider the underlying model as an oracle. This means that the method queries the trained model by inputting an evaluation dataset and registering the predictions made on each instance.
3. The output ruleset must be comprised of if-then rules or rules that can be translated into this format.

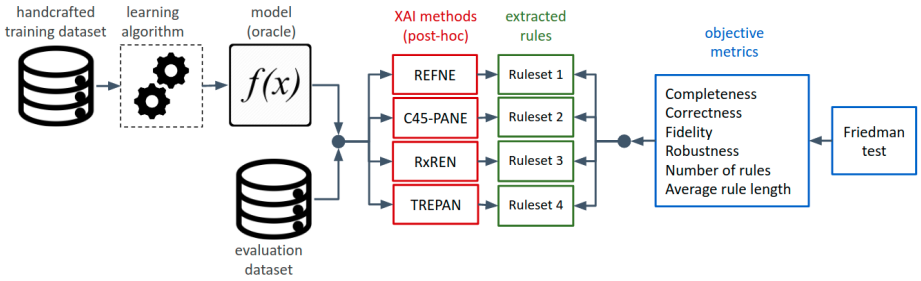

Fig. 1: Diagrammatic view of the design of the experiment carried out in this study for evaluating and comparing the degree of explainability of rulesets, automatically extracted by four methods from machine learned models.

Four rule-extraction methods fulfil these criteria, namely C4.5Rule-PANE, REFNE, RxREN and TREPAN. Their algorithms are summarised as pseudo-codes in Table 5 in the appendix and briefly described in Section 2. These methods were tested on eight public datasets (listed in Table 1) retrieved from the UCI Machine Learning Repository[1]. The datasets were selected according to the

[1] https://archive.ics.uci.edu/ml/index.php

following criteria: (I) all the features must be manually engineered by humans, (II) they must contain enough instances to avoid the curse of dimensionality, meaning too many features for too few instances, (II) the dependent variable is categorical whereas the independent variables are both continuous and categorical predictors, and (III) the number of instances is in the order of thousands (thus also limiting the number of features).

Table 1: Properties of the selected datasets.

Dataset	Total instances	Training instances	No. of input features	No. of continuous (categorical) features	No. of classes
Abalone	4177	2924	8	7 (1)	29
Contraceptive	1473	1031	10	2 (8)	3
Credit Germany	1000	700	20	7 (13)	2
Mushroom	8124	5687	22	0 (22)	2
Page-Block	5473	3831	9	9 (0)	5
Wave Form	5000	3500	40	40 (0)	3
Wine Quality	6497	4548	11	11 (0)	7
Yeast	1484	1039	8	8 (0)	10

Six metrics were selected to assess, in a objective and quantitative manner, the degree of explainability of the rulesets generated by the XAI methods from neural networks trained on the eight datasets. The objectivity is reached by excluding any human intervention or expert's domain knowledge in this evaluation process. Two of these metrics, number of rules and average rule length, are attributes of explainability. The other four, completeness, correctness, fidelity and robustness, are general validation factors to be fulfilled by any method for explainability. Table 3 reports their definition and the formulas used to calculate them. The ideal ruleset should minimise both the number of rules and the average rule length in order to be easily interpreted and understood by end-users [16]. In contrast, a ruleset must score high in terms of the other four metrics. This means that it can appropriately classify any input instances, it is faithful to the underlying model and its inferences do not vary when inputs are slightly distorted by applying a gaussian noise. The models trained on the eight datasets were all vanilla feed-forward neural networks with a single hidden layer. These simple networks were chosen to assess the feasibility of the proposed experiment. The number of nodes in the hidden layer, together with other hyperparameters of the networks, was determined by performing a grid search to guarantee the absence of overfitting/underfitting and reach the highest feasible prediction accuracy. Table 2 reports the list of the optimal values of the hyperparameters together with the accuracies obtained on the eight datasets.

To answer the research question, the experiment was designed as shown in the diagram of Figure 1. A model, which is the output of a learning algorithm (in this study vanilla feed-forward neural networks) trained on an input dataset, and an evaluation dataset were fed into the four XAI methods under analysis. Each method extracted a set of if-then rules whose degree of explainability was assessed with six objective and quantitative metrics. This process was repeated over the eight datasets and their neural networks.

Table 2: Optimal hyperparameters of neural networks obtained through grid search procedure, grouped by dataset, and their resulting accuracies.

Model parameters	Dataset list							
	Abalone	Contrac.	Credit Germany	Mushroom	Page Block	Wave Form	Wine Quality	Yeast
Optimizer	NAdam	NAdam	Adam	NAdam	NAdam	Adagrad	NAdam	NAdam
Weight initialisation	Glorot Normal	Uniform	Uniform	Glorot Uniform	He-Uniform	He-Uniform	He-Normal	He-Normal
Activation f.	Relu	Tanh	Softplus	H. Sigm.	Tanh	Linear	Softsign	Tanh
Dropout rate	30%	10%	0%	0%	0%	0%	30%	10%
Weight const.	4	5	3	4	5	1	2	3
Batch size	80	20	40	20	20	20	40	10
Epochs	100	50	100	100	100	50	50	100
Hidden neur.	30	30	25	15	15	10	20	15
Accuracy (test set)	36.88% (26.22%)	57.42% 54.81%	74.86% (73.39%)	100% (100%)	96.36% (95.46%)	87.17% (85.83%)	51.90% (53.02%)	62.43% (56.95%)

Table 3: Objective metrics to assess the explainability of rulesets.

Factor	Definition	Formula
Completeness	Ratio of input instances covered by rules (c) over total input instances (N) [8]	$\frac{c}{N}$
Correctness	Ratio of input instances correctly classified by rules (r) over total input instances [15]	$\frac{r}{N}$
Fidelity	Ratio of input instances on which the predictions of model and rules agree (f) over total instances [22]	$\frac{f}{N}$
Robustness	The persistence of methods to withstand small perturbations of the input (δ) that do not change the prediction of the model ($f(x_n)$) [3, 18]	$\frac{\sum_{n=1}^{N} f(x_n) - f(x_n + \delta)}{N}$
Number of rules	The cardinality of the ruleset (A) generated by the four methods under analysis [11, 13, 16]	$\|A\|$
Average rule length	The average number of antecedents, connected with the AND operator, of the rules contained in each ruleset [24]. a_i represents the number of antecedents of the i^{th} rule and R the number of rules.	$\frac{\sum_{i=1}^{R} a_i}{R}$

Each dataset was split into a training (70%) and an evaluation (30%) datasets, so the majority of the data was used to train the model whilst leaving enough data in the evaluation dataset to guarantee the representation of all the output classes. The final step of this experiment consists of ranking, in an objective and automatic way, the selected XAI methods according to these six metrics. The research hypothesis, to be verified with a statistical test, is that there are statistically significant differences in the degree of explainability of rulesets automatically extracted by the four methods. The Friedman test, a non-parametric statistical test designed to detect differences in treatments (the four XAI methods) across multiple test attempts (the six metrics), was applied to check whether any methods ranked consistently higher (or lower) according to the metrics of choice on each dataset. The alternative hypothesis of the test is that there are significant differences in the results of the four methods, hence one of them can be ranked as the best. The Friedman test was chosen instead of ANOVA because it was not possible to fulfil the assumption of the latter on the distribution of the samples that must come from normally distributed populations with equal standard deviations.

4 Results and discussion

The results obtained from this experiment are reported in Figures 2 and 3. Noticeably, all the methods but REFNE produced rulesets that reach 100% of completeness, meaning that the rules cover all the instances of the training and test datasets. These three methods use different strategies to achieve this result, as explained in Section 2. It is worth pointing out that all the four methods achieved more than 80% fidelity and robustness on the Credit Germany dataset, but these results are flawed. The 70% of the instances included in this dataset belong to the same output class: a good credit rating. The neural network trained on it assigned every new input instance to the majority class, thus completely ignoring the alternative class (a bad credit rate).

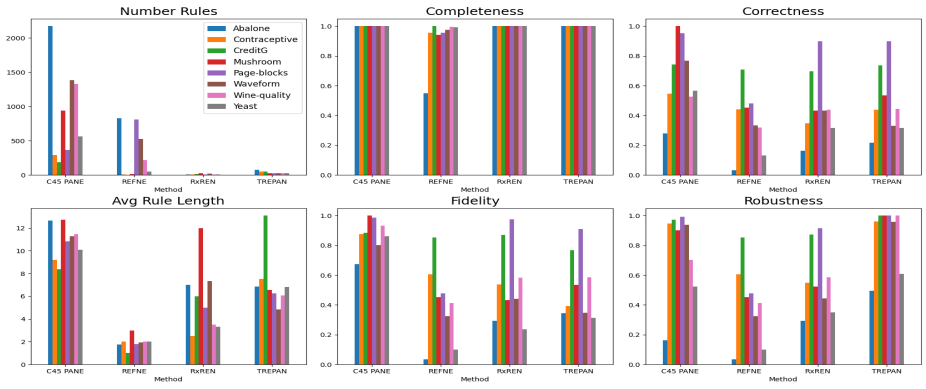

Fig. 2: Quantitative measures of the degree of explainability of the rulesets automatically generated by four rule-extraction methods, grouped by method.

As shown in Table 2, both the test and train accuracy of this network are respectively 74.9% and 73.4%, corresponding almost precisely to the portion of the input instances belonging to the majority class. The four rule-extraction methods captured this behaviour and they returned rulesets whose prediction is always "good-credit-rating". This issue is also the reason why the four methods reach the same level of correctness on the Credit Germany datasets (around 70%, in line with the underlying network) whereas they present differences in the results related to the other seven datasets. Overall, C4.5Rule-PANE performed better than the other three methods in terms of correctness, fidelity and robustness. However, the charts related to the number of rules and their average length suggest some drawbacks to reach these results. C4.5Rule-PANE indeed produced the biggest rulesets across all datasets under analysis, compared to the other three methods. This hampers the interpretability of its rulesets. REFNE created instead the smallest rulesets in terms of the average number of antecedents because of its algorithms limits to three the antecedents for each rule.

However, it has the downside of generating many rules, being the method that can be ranked second worst according to this metric. In contrast, RxREN and TREPAN produced rulesets with a few numbers of rules, ranging between 5 and 76, but their average length is comparable to C4.5Rule-PANE, especially on the Mushroom and Credit Germany datasets. RxREN and TREPAN reached also similar results in terms of fidelity and correctness, but TREPAN significantly outperformed all the other methods in robustness. Likely, this is because its algorithm generates rules based on binary splits of the datasets, thus making it insensitive to small variations in the data. Apparently, REFNE is the method that can be ranked as the worst across all the metrics but rule length. This is the consequence of extending the original training dataset with randomly generated data which breaks the relationships that might exist among predictors.

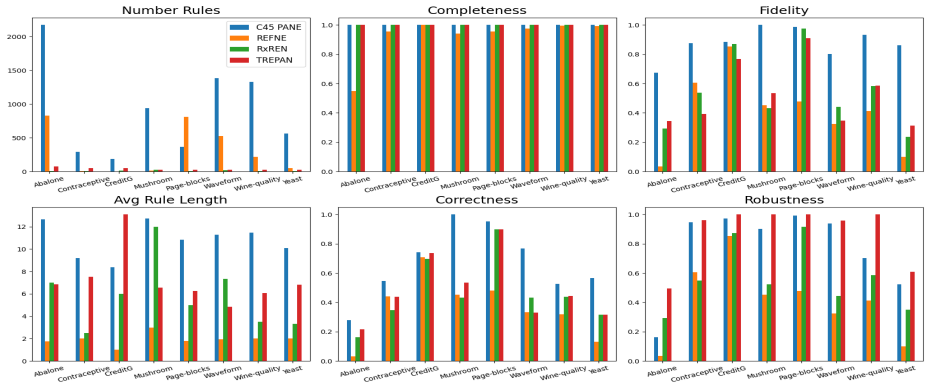

Fig. 3: Quantitative measures of the degree of explainability of the rulesets automatically generated by four rule-extraction methods, grouped by dataset.

Abalone is the dataset that pushes all the four XAI methods towards their limits as they all reach poor explainability on it. Indeed, this is the dataset with the lowest levels of correctness, fidelity and robustness across all the four methods. Meanwhile, it is also the dataset with the biggest ruleset per number of rules produced with C4.5Rule-PANE, REFNE and TREPAN. Only RxREN managed to keep this metric in line with the other seven datasets. Abalone is also among the datasets with the highest average rule length. Likely, this happened because it has by far the highest number of output classes (29) and seven out of its eight features are continuous. Furthermore, the neural network trained on Abalone reaches low prediction accuracy on both the training and test datasets, meaning that both the neural network and the four methods struggle in identifying which relationships among the data are decisive for correctly predicting the output classes. This might hint that these methods are not adapt to generate interpretable rulesets to explain predictive models trained on complex datasets, made of diverse and dispersed data.

In summary, this experiment provided a few interesting insights. Firstly, the results suggest that there is a trade-off between the size of the rulesets, in terms of both the number of rules and antecedents and the other four metrics, namely completeness, correctness, fidelity and robustness. In other words, when a method for rule extraction produces small rulesets or short rules, then the latter four metrics tend to score low. Secondly, all the methods captured the behaviour of the neural network trained on the Credit Germany dataset, which was highly unbalanced, as it ignored the minority output class by assigning almost every input instance to the majority class. Someone can argue this was expected because no up or down-sampling was applied to the majority or minority classes nor training with stratification was performed. However, this is a promising result as it shows that the selected XAI methods are actually capable to correctly uncover this situation by producing rulesets that minimises the latter four metrics. Finally, the Friedman test was applied to check whether these differences are statistically significant and a method can be considered superior to the others according to the six metrics under analysis across the selected datasets. The test output statistics and p-values are reported in Table 4 and Figure 4. All the p-values are lower than the typical tolerance level of 5%, thus there is strong evidence in support of the alternative hypothesis. This means that one of the four methods perform consistently better than the others, as shown in the last row of Table 4. Specifically, C4.5Rule-PANE is consistently ranked higher than the other methods across datasets with TREPAN coming as the second one. It seems that the mechanisms for splitting the space determined by the test set (normalised information gain and the gain ratio criterion), are more suitable than the mechanisms followed by the other two methods, respectively checking if there are values such that all the instances possessing them fall into the same class (REFNE) and determining the data ranges that minimise the number of misclassified instances.

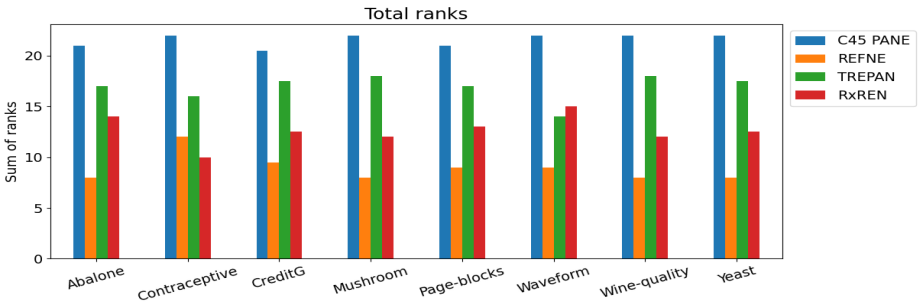

Fig. 4: Ranks of the four rule-extraction methods, grouped by datasets.

Table 4: Output statistics and p-values of the Friedman test, grouped by dataset.

	Dataset list							
	Abalone	Contrac.	Credit Germany	Mushroom	Page Block	Wave Form	Wine Quality	Yeast
Test statistic	9.643	9.000	8.760	12.429	8.571	9.214	12.429	12.055
p-value	0.022	0.029	0.033	0.006	0.036	0.027	0.006	0.007
Superior	C45-Pane	C45-Pane	C45-Pane	C45-Pane	C45-Pane	C45-Pane	C45-Pane	C45-Pane

5 Conclusions

This study presented a novel approach to evaluate and compare four XAI methods which extract rules from black-box machine-learned models, trained via vanilla neural networks on eight datasets. Six objective metrics were identified in the literature, namely ruleset cardinality, number of antecedents, completeness, fidelity, correctness, robustness. The Friedman test was used to check if one of the selected methods ranked consistently higher then the others across these metrics. The experiment provided sufficient evidence to support the alternate hypothesis of the Friedman test, hence one of the methods, specifically the C45-Pane, based on a feature splitting algorithm, outperformed the others. Furthermore, the selected metrics proved to be apt to highlight the weaknesses and strengths of the tested methods, thus providing scholars with an approach to test their XAI methods. Future work will extend this research study with additional metrics, datasets, deep neural networks and by performing feature pre-processing and cross-validation on the input datasets. This should enhance the intelligibility of the trained models and avoid issues during their training process due to unbalances in the data distribution among the output classes. It is also worth investigating, with a human-in-the-loop approach, the correlation of these metrics against qualitative perceptions gathered from humans.

References

1. Abdul, A., Vermeulen, J., Wang, D., Lim, B.Y., Kankanhalli, M.: Trends and trajectories for explainable, accountable and intelligible systems: An hci research agenda. In: Proceedings of the CHI Conference on Human Factors in Computing Systems. pp. 582–599. ACM, Montréal, Canada (2018)
2. Alonso, J.M., Castiello, C., Mencar, C.: A bibliometric analysis of the explainable artificial intelligence research field. In: International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems. pp. 3–15. Springer, Cádiz, Spain (2018)
3. Alvarez-Melis, D., Jaakkola, T.S.: On the robustness of interpretability methods. In: Proceedings of the 2018 ICML Workshop in Human Interpretability in Machine Learning. pp. 66–71. ICML, Stockholm, Sweden (2018)
4. Augasta, M.G., Kathirvalavakumar, T.: Reverse engineering the neural networks for rule extraction in classification problems. Neural processing letters **35**(2), 131–150 (2012)
5. Besold, T.R., Kühnberger, K.U.: Towards integrated neural–symbolic systems for human-level ai: Two research programs helping to bridge the gaps. Biologically Inspired Cognitive Architectures **14**, 97–110 (2015)

6. Craven, M., Shavlik, J.W.: Extracting tree-structured representations of trained networks. In: Advances in neural information processing systems. pp. 24–30. MIT Press, Denver, Colorado, USA (1996)
7. Craven, M.W., Shavlik, J.W.: Using sampling and queries to extract rules from trained neural networks. In: Machine Learning Proceedings, pp. 37–45. Elsevier, New Brunswick, New Jersey, USA (1994)
8. Cui, X., Lee, J.M., Hsieh, J.: An integrative 3c evaluation framework for explainable artificial intelligence. In: AI and semantic technologies for intelligent information systems (SIGODIS). pp. 1–10. AIS eLibrary, Cancún, Mexico (2019)
9. Došilović, F.K., Brčić, M., Hlupić, N.: Explainable artificial intelligence: A survey. In: 41st Int. Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO). pp. 0210–0215. IEEE, Opatija, Croatia (2018)
10. Ferri, C., Hernández-Orallo, J., Ramírez-Quintana, M.J.: From ensemble methods to comprehensible models. In: International Conference on Discovery Science. pp. 165–177. Springer, Lübeck, Germany (2002)
11. Freitas, A.A.: On rule interestingness measures. In: Research and Development in Expert Systems XV, pp. 147–158. Springer, United Kingdom (1999)
12. Freitas, A.A.: Are we really discovering interesting knowledge from data. Expert Update (the BCS-SGAI magazine) **9**(1), 41–47 (2006)
13. García, S., Fernández, A., Luengo, J., Herrera, F.: A study of statistical techniques and performance measures for genetics-based machine learning: accuracy and interpretability. Soft Computing **13**(10), 959 (2009)
14. Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., Pedreschi, D.: A survey of methods for explaining black box models. ACM computing surveys (CSUR) **51**(5), 93:1–93:42 (2018)
15. Ignatiev, A.: Towards trustable explainable ai. In: Proceedings of the 29th Int. Joint Conference on Artificial Intelligence. pp. 5154–5158. Yokohama, Japan (2020)
16. Lakkaraju, H., Bach, S.H., Leskovec, J.: Interpretable decision sets: A joint framework for description and prediction. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining. pp. 1675–1684. ACM, San Francisco, California, USA (2016)
17. Letham, B., Rudin, C., McCormick, T.H., Madigan, D., et al.: Interpretable classifiers using rules and bayesian analysis: Building a better stroke prediction model. The Annals of Applied Statistics **9**(3), 1350–1371 (2015)
18. Liu, S., Wang, X., Liu, M., Zhu, J.: Towards better analysis of machine learning models: A visual analytics perspective. Visual Informatics **1**(1), 48–56 (2017)
19. Longo, L., Goebel, R., Lecue, F., Kieseberg, P., Holzinger, A.: Explainable artificial intelligence: Concepts, applications, research challenges and visions. In: International Cross-Domain Conference for Machine Learning and Knowledge Extraction. pp. 1–16. Springer, Dublin, Ireland (2020)
20. Miller, T.: Explanation in artificial intelligence: insights from the social sciences. Artificial Intelligence **267**, 1–38 (2019)
21. Rizzo, L., Longo, L.: A qualitative investigation of the explainability of defeasible argumentation and non-monotonic fuzzy reasoning. In: Proceedings for the 26th AIAI Irish Conference on Artificial Intelligence and Cognitive Science Trinity College Dublin. pp. 138–149. CEUR-WS.org, Dublin, Ireland (2018)
22. Saad, E.W., Wunsch II, D.C.: Neural network explanation using inversion. Neural networks **20**(1), 78–93 (2007)
23. Vilone, G., Longo, L.: Explainable artificial intelligence: a systematic review. arXiv preprint arXiv:2006.00093 (2020)

24. Wu, M., Hughes, M.C., Parbhoo, S., Zazzi, M., Roth, V., Doshi-Velez, F.: Beyond sparsity: Tree regularization of deep models for interpretability. In: Thirty-Second AAAI Conference on Artificial Intelligence. pp. 1670–1678. AAAI Press, New Orleans, Louisiana, USA (2018)
25. Zhou, Z.H., Jiang, Y.: Medical diagnosis with c4. 5 rule preceded by artificial neural network ensemble. IEEE Transactions on information Technology in Biomedicine **7**(1), 37–42 (2003)
26. Zhou, Z.H., Jiang, Y., Chen, S.F.: Extracting symbolic rules from trained neural network ensembles. AI Communications **16**(1), 3–15 (2003)

A Appendix

Table 5: Pseudo-code of the algorithms of four rule-extraction methods.

```
1: REFNE(X):
2:     R = empty ruleset
3:     Create synthetic dataset S by varying
           each input feature of X across
           its value range
4:     y' = Oracle(model, S)
5:     Select a categorical feature F of S
6:     Find value U such that all y'
           with U belong to class C
7:     Create rule r
8:     If fidelity of r > delta:
9:         Add r to ruleset R
10:        Remove y' covered by r from S
11:        If size(S) = 0 return R
12:        Else select another F
13:    If all F have been selected, discretize
           continuous variables with ChiMerge
```

```
1: C4.5Rule-PANE(X):
2:     y' = Oracle(model, X)
3:     Create synthetic dataset S by
           varying each input feature of
           X across its value range
4:     y'' = Oracle(model, S)
5:     xSynth = concatenate X and S
6:     ySynth = concatenate y' and y''
7:     C45_build_tree(xSynth, ySynth)
```

```
1: TREPAN(training_examples, features):
2:     Queue = 0
3:     For each example E in training_examples:
           class_E = Oracle(model, E)
4:     Initialize the root of tree T as leaf
5:     Put (T,training_examples,{}) into Queue
6:     While size(Queue) > 0 &
7:         size(T) < tree_limit:
8:             Remove node N from head of Queue
9:             example_N = examples stored w/ N
10:            constraint_N = constraints
                       stored with N
11:            Use features and example_N to
                   build set of candidate splits
12:            Oracle(constraint_N, example_N)
                   to evaluate splits
13:            S = best binary split
14:            Search for best m-of-n split S'
                   using S as seed
15:            Make N an internal node with
                   split S'
16:            Put (N,example_N,constraint_N,S')
                   into Queue
```

```
1: RxREN(X):
2:     T = set of correctly classified
           instances of X
3:     original_acc = model accuracy
4:     Remove each input feature and
           estimate new accuracy n_acc
5:     If n_acc > original_acc - 1%, then
           prune feature
6:     E = instances from T incorrectly
           classified by pruned network
7:     Compute mandatory data ranges for
           each significant feature from E
8:     Construct rules for each class using
           mandatory data range
9:     Check if each new rule improve the
           accuracy of ruleset
10:    Classify test examples using ruleset
11:    Find min and max of misclassified
           examples corresponding
           to each class
12:    Replace previous data ranges if new
           min and max improves accuracy
           of ruleset
```