The PSyKE Technology for Trustworthy Artificial Intelligence

The PSyKE technology for Trustworthy Artificial Intelligence*

Roberta Calegari[1][0000−0003−3794−2942] and Federico Sabbatini[2][0000−0002−0532−6777]**

[1] Alma AI – Alma Mater Research Institute for Human-Centered Artificial Intelligence, Alma Mater Studiorum—Università di Bologna, Italy
roberta.calegari@unibo.it
[2] Department of Pure and Applied Sciences (DiSPeA)
University of Urbino, Via S. Chiara, 27, 61029, Urbino, Italy
f.sabbatini1@campus.uniurb.it

Abstract. Transparency is one of the "Ethical Principles in the Context of AI Systems" as described in the Ethics Guidelines for Trustworthy Artificial Intelligence (TAI). It is closely linked to four other principles – respect for human autonomy, prevention of harm, traceability and explainability – and involves numerous ways in which opaqueness can have undesirable impacts, such as discrimination, inequality, segregation, marginalisation, and manipulation. The opaqueness of many AI tools and the inability to understand the underpinning black boxes contradicts these principles as well as prevents people from fully trusting them. In this paper we discuss the PSyKE technology, a platform providing general-purpose support to symbolic knowledge extraction from different sorts of black-box predictors via many extraction algorithms. The extracted knowledge results are easily injectable into existing AI assets making them meet the transparency TAI requirement.

Keywords: Trustworthy Artificial Intelligence · Transparency · Explainability · Symbolic knowledge extraction · PSyKE

1 Introduction

The innovative potential of Artificial Intelligence (AI) is clear, but AI tools can reflect, amplify, and even create untrustworthy behaviours, beliefs, decisions or results [15]. As we use AI systems to formalise, scale, and accelerate processes, we have the opportunity, as well as the duty, to revise and enhance the existing processes, avoiding perpetuating existing patterns of untrustworthiness, by detecting, diagnosing, and repairing them. To trust these systems, domain experts and stakeholders need to trust the decisions made by them. Europe's strategy

* This work has been partially supported by the EU ICT-48 2020 project TAILOR (No. 952215) and by the European Union's Horizon 2020 research and innovation programme under G.A. no. 101017142 (StairwAI project).

** Corresponding author

aims to create an AI Ecosystem of Excellence and Trust where ethical and legal principles are pursued in all AI systems. Transparency is one of the "Ethical Principles in the Context of AI Systems" as described in the Ethics Guidelines for Trustworthy Artificial Intelligence (EGTAI) [9] and in the first AI regulation (the "AI Act") [8]. It is closely linked to four other principles (respect for human autonomy, prevention of harm, traceability and explainability) and involves numerous ways in which opaqueness can have undesirable impacts, such as discrimination, inequality, exclusion, segregation, marginalisation, exploitation, and manipulation.

However, the translation of ethical principles and EGTAI into practical requirements are needed to boost high quality AI innovation in Europe. Concrete methods to ensure that AI systems adhere to the transparency requirement can be borrowed from the explainability domain, since providing explanations concurs to achieve transparency. Different strategies can be exploited to meet transparency and explainability [11]. For instance, it is possible to obtain explainable data-driven solutions *only* by using *interpretable* algorithms [16]—such as decision lists, decision trees and sparse integer linear models, and algorithms based on discrete optimisation. However, this kind of technique often has repercussions on the final predictive performance, since most effective algorithms – like artificial neural networks – are not taken into account. Deriving *post-hoc* explanations [14] is an alternative strategy aimed at reverse-engineering the black-box (BB) inner behaviour to make it explicit. This is a way of combining the performance of prediction-effective (even if opaque) machine learning models with human-interpretable output predictions.

Symbolic knowledge extraction (SKE) represents one of the most promising techniques to derive *post-hoc* explanations from sub-symbolic BB models and interpret the notion of explainability under the transparency perspective, i.e. proposing a transparent model adhering to the not transparent predictor. Its main idea is to build a *symbolic* – and thus interpretable – model that mimics the behaviour of the original BB, intended as the capability to provide outputs that are as close as possible w.r.t. those of the underlying BB queried on the same inputs. Symbols may consist of comprehensible knowledge—e.g., lists or trees of *rules* that can be exploited to either derive predictions or to better understand the BB behaviour and, as a further step, as knowledge on which to perform any kind of logical reasoning. Currently, SKE techniques have been already applied in a wide variety of areas, ranging from medical diagnosis [10] to finance [1] and astrophysics [22]. Despite the wide adoption of SKE and the existence of different techniques for extracting symbolic knowledge out of a BB, a unified and general-purpose software technology supporting such methods and their comparison is currently lacking. In other words, the burden of implementing SKE algorithms, or selecting the best one from the state of the art, is currently on AI stakeholders alone, who are likely to realise custom solutions for a specific application need. Other than slowing down the adoption of SKE as an effective method for reaching transparency, such a lack of viable technologies is somewhat anachronistic in the data-driven AI era, where a plethora of libraries and frameworks

are flourishing, targeting all major programming paradigms and platforms, and making state-of-the-art machine learning (ML) algorithms easily accessible to the general public—cf. SciKit-Learn[1] for Python.

Accordingly, in this paper we present a general-purpose Platform for Symbolic Knowledge Extraction – PSyKE – as a way to practicalise the TAI requirement – transparency in particular – from high-level principles to concrete methods. Moreover, one of the PSyKE goals is filling the gap between the current state of the art of SKE and the available technology as well as providing a concrete toolkit for testing, evaluating and reaching transparency in AI applications. It provides a controlled experimentation environment for transparency via SKE methods enabling the creation of different simulations/experiments for the specific application at hand. The framework comes as a toolkit in which experiments on transparency can be built and run, comparing different solutions, and selecting the best option. More precisely, PSyKE is conceived as an open library where different sorts of knowledge extraction algorithms can be realised, exploited, or compared. PSyKE supports rule extraction from both classifiers and regressors, and makes the extraction procedure as transparent as possible w.r.t. the underlying BB, depending on the particular extraction procedure at hand. The extraction of first-order logic clauses is also supported, with the twofold advantage of providing human- and machine-interpretable rules as output. These can then be used as either an explanation for the original BB or as a starting point for further symbolic computations and reasonings.

2 The PSyKE framework

PSyKE[2] [18, 19] is a platform providing general-purpose support to symbolic knowledge extraction from different sorts of black-box predictors via many extraction algorithms.

2.1 Functionalities & main components

PSyKE comes as a software library providing general-purpose support to the extraction of logic rules out of BB predictors by letting users choose the most adequate SKE method for the task and data at hand. A unified API covering virtually all extraction algorithms targeting supervised learning tasks is exposed by the framework and experiments can also be run via a GUI. Currently, PSyKE grants access to state-of-the-art SKE algorithms providing the implementations of several interoperable, interchangeable, and comparable extraction SKE methods [6, 7, 13, 20, 2, 17]. PSyKE is conceived as an open-ended project, exploitable to design and implement new extraction procedures behind a unique API.

Essentially, PSyKE is designed around the notion of *extractor*, whose overall design is depicted in Figure 1. Within the scope of PSyKE, an extractor

[1] https://scikit-learn.org/stable
[2] https://apice.unibo.it/xwiki/bin/view/PSyKE/

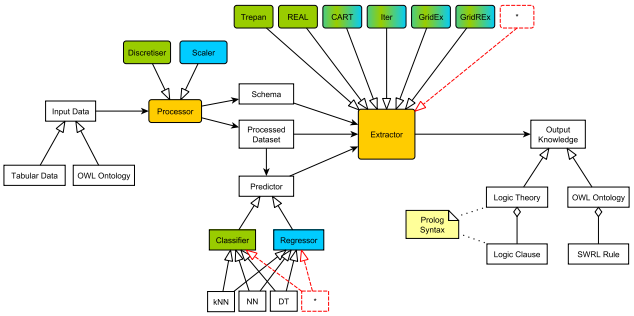

Fig. 1: PSyKE design

is any algorithm accepting a machine learning predictor as input (classifier or regressor), and producing a *theory* of *logic* rules as output.

PSyKE extractors require additional information to complete the extraction task. Such information consists of the data set used to train the predictor and its schema. Data sets are required to allow the extraction procedure to inspect the BB behaviour – and therefore build the corresponding output rules – whereas schemas are required to allow *(i)* the extraction procedure to take decisions based on feature *types*, and *(ii)* the extracted knowledge to be more interpretable by referring to the feature *names*. Accordingly, extractors expect also the data set and its schema metadata as input. Figure 1 shows also the *discretiser* and *scaler* components. The former aims at providing some facilities for discretising (binarising) data sets including continuous (categorical) data. This is a procedure often needed for data sets involving these kinds of attributes to be given as input to extractors only accepting discrete or binary input features.

2.2 Architecture & API

As depicted in Figure 2, a key role in the design of PSyKE is played by the `Extractor` interface, defining the general contract of any knowledge-extraction procedure. Each `Extractor` encapsulates a single machine learning `Predictor` and a particular `Discretisation` strategy. Given a set of inputs, an extractor is capable of extracting a `Theory` of logic `Rules` out of a `DataFrame`, containing the examples the `Predictor` has been trained upon.

PSyKE assumes underlying libraries to be available on the runtime adopted for implementation, from which AI facilities can be inherited. These include: a machine learning library, exposing *ad-hoc* types aimed at representing data sets, data schemas, or predictors, and a symbolic AI library, exposing *ad-hoc* types for representing and manipulating logic theories, clauses, and rules. PSyKE inherits high-level abstractions from these libraries. These include the following components:

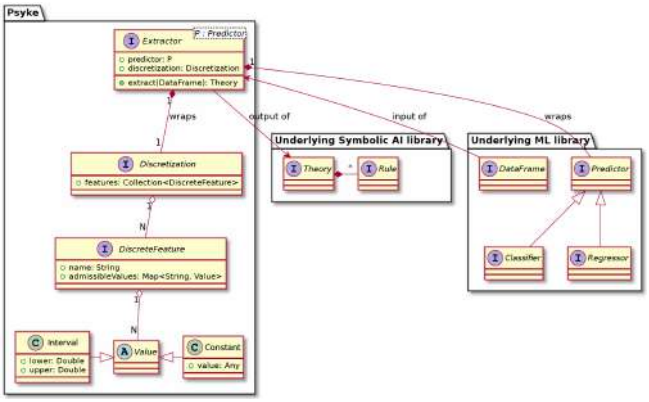

Fig. 2: PSyKE's `Extractor` interface

`DataFrame` — a container of tabular data, where rows commonly denote instances, and columns denote their features, while bulk operations are available to manipulate the table as a whole, as well as any row/column of its;
`Predictor<R>` — a computational entity which can be trained (a.k.a. fitted) against a `DataFrame` and used to draw predictions of type `R`;
`Classifier<R>` — a particular case of predictor where `R` represents a type having a finite amount of admissible values;
`Regressor<R>` — a particular case of predictor where `R` represents a type having a potentially infinite (possibly continuous) amount of admissible values;
`Rule` — a semantic, intelligible representation of the function mapping `Predictor`'s inputs into the corresponding outputs, for a portion of the input space;
`Theory` — an ordered collection of rules.

For example, PSyKE borrows ML-related abstractions – such as `DataFrame`, `Predictor`, or `Classifier` – from either Pandas or Scikit-Learn Python libraries. Similarly, it borrows high-level symbolic-AI-related abstractions – such as `Theory` or `Rule` – from 2P-Kt[3] [5].

PSyKE constructs its notion of `Extractor` upon these inherited concepts—thus designing an `Extractor` as any method capable of extracting logic `Rules` out of some trained `Predictor`. PSyKE extractors are bound to the particular underpinning black-box `Predictor`, as well as to the `Discretisation` strategy exploited for the input space. `Extractors` also expose a method for extracting an explainable `Theory` from the `Predictor` – namely, `extract` – and a method to draw predictions by using the extracted rules—namely, `predict`. Any attempt to use the extracted rules to draw explainable predictions triggers extraction first—i.e., the prediction procedure implies extraction. Both extraction and prediction rely on a `DataFrame` that must be provided by the user upon invocation. Extractors, in the general case, may also be used to perform rule induction from data, without any intermediate predictor.

[3] https://github.com/tuProlog/2ppy

It is worth noting that `Predictor`s are parametric types. The meta-parameter `R` represents the type of predictions the predictor may produce. The rules possibly extracted by such predictors – as well as the predictions extracted – may differ significantly depending on the particular data and on the selected predictors. For instance, when rules are extracted from mono-dimensional regressors, `R` may be the type of floating point numbers, whereas, for multi-class classifiers, `R` may consist of the set of types (like integer, string, ...). Depending on the nature of `R`, the extracted rules possibly differ significantly. However, the proposed API makes it possible to switch between different extraction algorithms and predictors with no changes in the PSyKE architecture.

Output rules produced by PSyKE's extractors may be more tailored on human-interpretability or agent-/machine-interoperability [21]. In the former case, a Prolog theory of logic clauses is provided as output. In the latter case, the knowledge is extracted as an OWL ontology containing SWRL rules.

3 Examples

In this section some examples showing PSyKE working in different scenarios are reported—i.e. the Iris data set[4] as classification task and the Combined Cycle Power Plant[5] (CCPP) data set as a regression case study.

3.1 Classification: the Iris data set

In the following we report the outcome of PSyKE when applying different SKE techniques to the Iris data set. All the results are resumed in Figure 3 and Table 1. Column "Predictor" represents the ML step of the process. Column "Extractor" represents the output of PSyKE. Different extraction procedures – namely, Iter, GridEx, and Cart – are applied to some selected BB classifiers. These predictors are a k-nearest neighbor with $k = 5$ (5-NN), a decision tree (DT) and a multilayer perceptron (MLP).

A numerical assessment of the aforementioned predictors and extractors is reported in Table 1 in terms of number of extracted rules and predictive performance w.r.t. data and BB predictions. The predictive performance is expressed through both classification accuracy and F_1 score metrics. Values are averaged upon 25 executions, each one with different random train/test splits, but the same test set percentage and same parameters for predictors and extractors. Table 1 also reports the underpinning BB predictor accuracy and the fidelity and accuracy of the extraction procedure.

It is worth noting that different SKE techniques can be easily compared and the best option for the scenario at hand can be selected thanks to the controlled experimentation environment provided by PSyKE.

[4] https://archive.ics.uci.edu/ml/datasets/iris
[5] https://archive.ics.uci.edu/ml/datasets/combined+cycle+power+plant

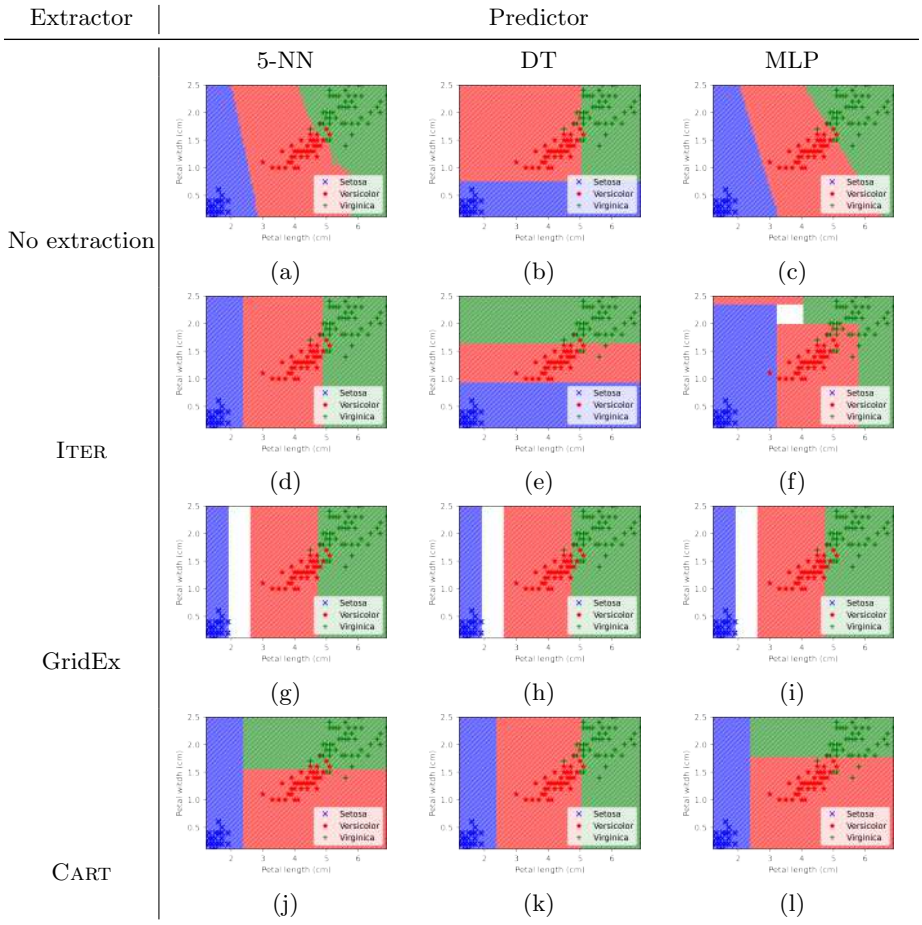

Fig. 3: Comparison between Iris data set input space partitionings performed by the algorithms implemented in PSyKE. Only the two most relevant features are reported—i.e., petal width and length.

3.2 Regression: the Combined Cycle Power Plant data set

In this example, PSyKE is exploited to extract rules out of different BB regressors trained upon the CCPP data set. The data set contains 9568 instances, each one composed of 4 real-valued input attributes.

Diverse regressors are trained on the CCPP data set: a 3-NN, a DT and a linear regressor (LR). Same as the previous example, PSyKE is used to extract logic rules out of the selected BB models exploring some of the SKE methods it supports—namely, Iter, GridEx, GridREx and Cart. Metrics for measuring the fidelity of the extractor w.r.t. the underlying BB predictions as well as the predictive accuracy w.r.t. the data are the mean absolute error (MAE) and R^2 score. The same metrics are used to assess the predictive performance of the BBs

Table 1: Comparison between predictive performance and fidelity measurements applied to the Iris data set. The best extractors are highlighted.

Predictor			**Extractor**					
Type	**Accuracy**	**F_1 score**	**Algorithm**	**Rules**	**Accuracy (data)**	**Accuracy (BB)**	**F_1 score (data)**	**F_1 score (BB)**
5-NN	0.96	0.96	Iter	3	0.91	0.93	0.91	0.93
			GridEx	**3**	**0.94**	**0.96**	**0.94**	**0.96**
			Cart	3	0.92	0.93	0.92	0.93
DT	0.96	0.96	**Iter**	**3**	**0.96**	**0.94**	**0.96**	**0.94**
			GridEx	**3**	**0.94**	**0.96**	**0.94**	**0.96**
			Cart	3	0.89	0.93	0.89	0.93
MLP	0.99	0.99	Iter	5	0.80	0.79	0.78	0.76
			GridEx	**3**	**0.94**	**0.96**	**0.94**	**0.96**
			Cart	3	0.95	0.93	0.95	0.93

and as for the Iris case study the extracted knowledge readability is expressed as number of rules.

The results of PSyKE applied to the CCPP data set are summarised in Figure 4 and Table 2. Each one of the extraction procedures suitable for regression tasks is applied to all the aforementioned BB regressors.

Figure 4 shows that all the extractors are able to capture the behaviour of the output values w.r.t. the input variables.

Table 2 reports the predictive performance of predictors and extractors. Values are averaged upon 25 executions, each one with different train/test splits, but with the same parameters for both predictors and extractors. Results show that in the case at hand all predictors have comparable performance in terms of MAE and R^2 score. Conversely, it is possible to notice that Cart, GridEx and GridREx always appear more explainable than Iter in terms of the number of extracted rules. From the Table it may be easily noticed also that GridEx and Cart generally present analogous performance. This fact depends on the nature of the corresponding output rules. Indeed, they both produce rules having constant output values, introducing an undesired discretisation of the predicted variable. Both of them are able to outperform Iter also in terms of predictive performance (smaller MAE and larger R^2 score).

On the other hand, GridREx outperforms all the other algorithms, achieving higher fidelity and readability. This depends on the regressive nature of its outputs, enabling the creation of more concise output rules performing more accurate predictions. Indeed, GridREx rules have as postconditions linear combinations of the input variables.

The nature of the different predictors and extractors used in this case study may be easily noticed in Figure 4. The boundaries identified by the 3-NN clearly follow a proximity pattern. Conversely, the DT performs variable slicing along each input dimension and the LR produces a gradual output value decrement

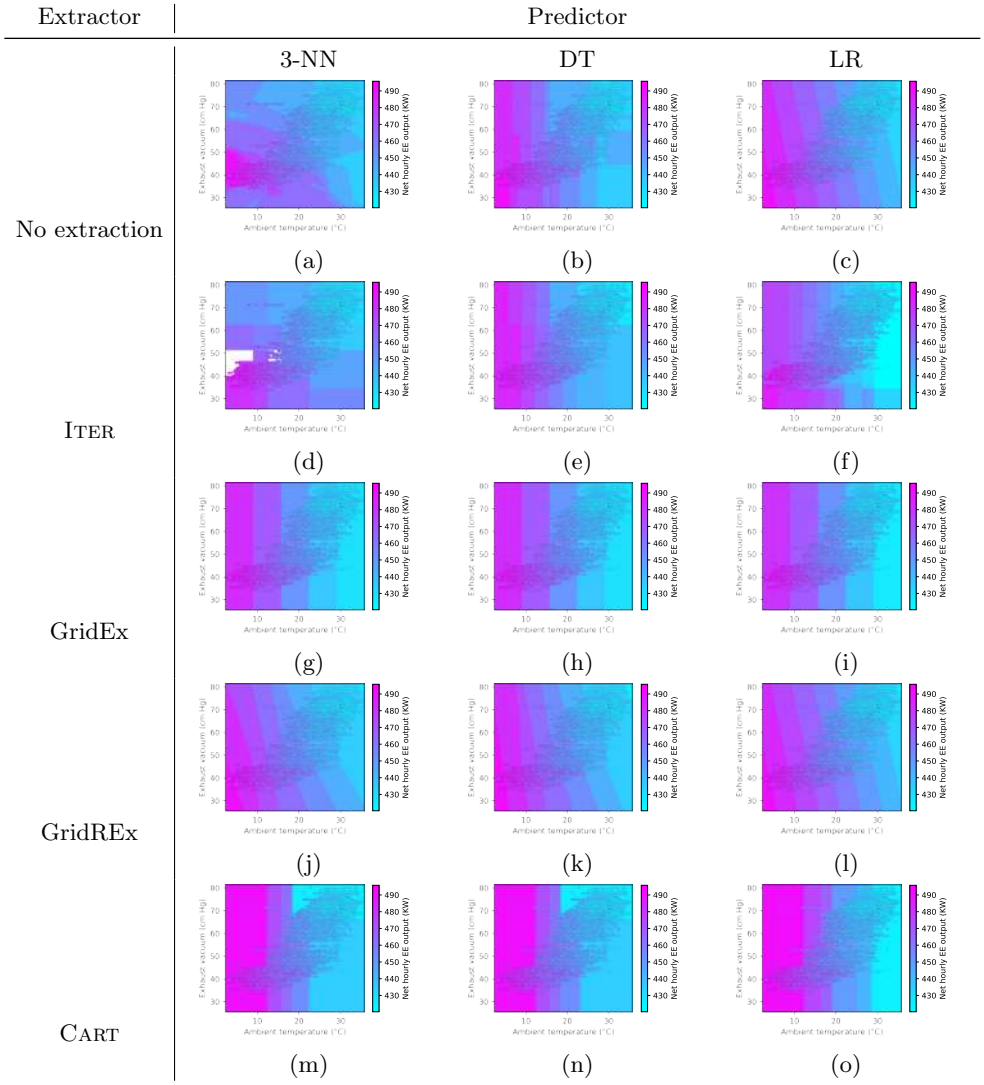

Fig. 4: Comparison between CCPP data set output predictions provided by the algorithms implemented in PSyKE. Only the two most relevant features are reported—i.e., ambient temperature and exhaust vacuum.

for growing input values. As for the extractors, for Cart the same considerations made for the DT hold. The hypercubic nature of Iter and GridEx is detectable by observing the rectangular boundaries provided by them. Finally, GridREx provides local linear regressive laws for hypercubic regions, merging the advantages of both DTs and LRs.

Table 2: Comparison between predictive performance and fidelity measurements applied to the CCPP data set. The number of extracted rules is also reported. The best extractors are highlighted.

Predictor			**Extractor**					
Type	**MAE**	**R^2 score**	**Algorithm**	**Rules**	**MAE**		**R^2 score**	
					(data)	**(BB)**	**(data)**	**(BB)**
3-NN	3.09	0.94	Iter	22	4.19	3.78	0.94	0.96
			GridEx	5	5.02	4.63	0.87	0.88
			GridREx	**5**	**3.25**	**2.52**	**0.94**	**0.96**
			Cart	6	4.45	3.90	0.89	0.91
DT	3.31	0.92	Iter	14	4.27	4.32	0.93	0.92
			GridEx	5	5.02	5.10	0.87	0.86
			GridREx	**5**	**3.24**	**3.38**	**0.94**	**0.93**
			Cart	6	4.46	4.50	0.89	0.88
LR	3.59	0.92	Iter	43	4.42	2.74	0.93	1.00
			GridEx	5	5.15	3.80	0.86	0.92
			GridREx	**1**	**3.59**	**0.00**	**0.93**	**1.00**
			Cart	6	4.97	3.49	0.87	0.93

Once again it is worth noting how PSyKE technology enables different SKE techniques to be compared. Such a comparison provide also a measure in terms of explainability and transparency that can be achieved out of the BB predictor.

3.3 PSyKE GUI

Figure 5 shows an example of PSyKE GUI screenshot in order to highlight how the toolkit also enables achieving fast and easy interactions with users. The GUI is simple and user-friendly, divided into 4 panels. The top panel is dedicated to the task selection (classification vs. regression) and to data set selection/pre-processing. Users can choose between several pre-defined data sets, as well as load a custom file. Furthermore, they can choose to discretise/scale the features and, on the right, it is possible to select among all the available features *(i)* the one to be used as output; *(ii)* those to be used as inputs; and *(iii)* those to be neglected. On the same panel it is possible to select two input features to be plotted together with the output feature. Plots appear in the right-most central panel of the GUI. The first one represents the data set instances, the second depicts the decision boundaries of the trained BB predictor and the third does the same for the selected extractor. Plots are shown after the proper button pressing, but each plot depends on the previous operations performed by the users. The predictor plot requires a BB predictor to be previously chosen and trained. This can be done by acting on the left-most central panel of the interface. Several models are available, each one with corresponding text boxes to allow users customise the required hyper-parameters. Users can also choose the train-test splitting percentage. Each parameter has a default value, so user inputs

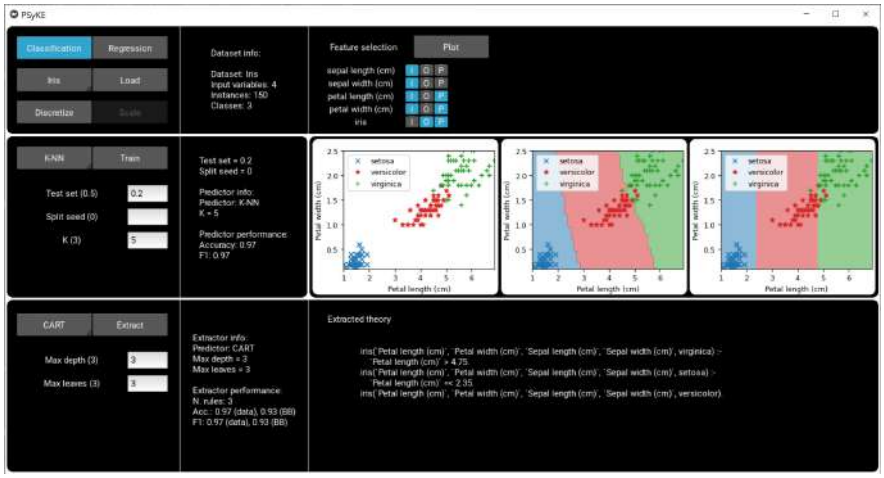

Fig. 5: PSyKE GUI

are optional. Analogously, the bottom-most panel is dedicated to the selection, training and tuning of knowledge extractors. Training an extractor enables the visualisation of the third plot.

The knowledge extracted with PSyKE extractors is displayed below the plots, in Prolog syntax. Finally, information about the chosen data set (e.g., number of features, classes and instances), predictor (e.g., parameters and predictive performance) and extractor (e.g., parameters, predictive performance and fidelity measurements) are shown next to the corresponding selection commands (after their selection).

The example reported in Figure 5 shows the application of PSyKE to the Iris data set. The data set has been loaded without discretisation and feature pruning, then a 5-NN has been trained on 80% of the data set. The CART extractor has finally been chosen, with maximum depth and maximum leaf amount equal to 3. Only input features about petal width and length have been selected to be plotted.

In conclusion, the framework provides the possibility to build different experiments in a controlled environment, enabling an easy exploitation of the technology and offering the possibility to compare the results in a simple way.

4 Impact

The PSyKE technology may impact many research areas. It provides a well-grounded technological basis and a software engineering practice for implementing/experimenting with the transparency and explainability dimensions in AI applications. It provides an extensible framework for collecting the SKE methods and approaches proposed in the literature, creating a controlled environment for testing, evaluating and comparing transparency. PSyKE has an important

role from the point of view of software engineering, providing a methodology that can be exploited for grounding all the TAI dimensions—i.e., the design and the implementation of a controlled experimentation environment that can act also as a sandbox for simulating the trustworthiness of an AI system. Accordingly, the framework provides a concrete example of the feasibility of building a practical toolkit for AI stakeholders to test the dimensions of TAI. Moreover, PSyKE has a role to play in the field of XAI [12]. Integrating symbolic and sub-symbolic AI – i.e., using them in synergy, as an ensemble – is a strategical research direction [4], and PSyKE offers a sound technological foundation for this purpose. Finally, the distributed systems community has the need for interoperable and general-purpose logic-based technologies that can be easy injectable into already existing systems [3]. There, PSyKE provides a technological layer easy injectable in distributed systems supporting agents' reasoning via the production of logical knowledge that can be exploited by agents.

Given all the potential of the described framework, there is room for several future research directions. PSyKE already enables the investigation of relevant research questions involving symbolic manipulation or automated reasoning, thanks to its modularity and interoperability. Under such a perspective, PSyKE enables exploring how to: *(i)* blend SKE with other AI techniques, and *(ii)* exploit SKE to build flexible intelligent systems.

Along these lines, future research directions will take into account the integration in the framework of a larger suite of methods for dealing with the most variety of datasets and predictors. Some preliminary experiments showed that the SKE algorithms can be exploited also for rule induction starting from data. This line is particularly interesting for all the cases in which a BB predictor is not available. Moreover, new SKE techniques are under development exploiting the combination of SKE with explainable clustering techniques increasing both performance and fidelity.

Finally, the framework is a preliminary example of how TAI dimensions can be tested and evaluated, and an interesting research line is to extend the environment in order to achieve a certification of the level of transparency – or more in general trustworthiness – for given AI applications. The challenge here is to find a way for defining effective metrics for the certification of TAI dimensions.

5 Conclusion

In this paper we discuss the PSyKE technology, a platform providing general-purpose support to symbolic knowledge extraction from different sorts of black-box predictors via many extraction algorithms, to be easily injectable into existing AI assets making them meet the transparency TAI requirement.

The framework provides a controlled experimentation environment in which transparency and explainability can be tested, assessed and compared. Even if still in a preliminary stage, it provides a software engineering practice for grounding all the TAI dimensions, translating them from high-level principles to practical requirements.

References

1. Baesens, B., Setiono, R., De Lille, V., Viaene, S., Vanthienen, J.: Building credit-risk evaluation expert systems using neural network rule extraction and decision tables. In: Storey, V.C., Sarkar, S., DeGross, J.I. (eds.) ICIS 2001 Proceedings. pp. 159–168. Association for Information Systems (2001), http://aisel.aisnet.org/icis2001/20
2. Breiman, L., Friedman, J., Stone, C.J., Olshen, R.A.: Classification and Regression Trees. CRC Press (1984)
3. Calegari, R., Ciatto, G., Mascardi, V., Omicini, A.: Logic-based technologies for multi-agent systems: A systematic literature review. Autonomous Agents and Multi-Agent Systems **35**(1), 1:1–1:67 (2021). https://doi.org/10.1007/s10458-020-09478-3, http://link.springer.com/10.1007/s10458-020-09478-3, collection "Current Trends in Research on Software Agents and Agent-Based Software Development"
4. Calegari, R., Ciatto, G., Omicini, A.: On the integration of symbolic and sub-symbolic techniques for XAI: A survey. Intelligenza Artificiale **14**(1), 7–32 (2020). https://doi.org/10.3233/IA-190036
5. Ciatto, G., Calegari, R., Omicini, A.: 2P-Kt: A logic-based ecosystem for symbolic AI. SoftwareX **16**, 100817:1–7 (Dec 2021). https://doi.org/10.1016/j.softx.2021.100817, https://www.sciencedirect.com/science/article/pii/S2352711021001126
6. Craven, M.W., Shavlik, J.W.: Using sampling and queries to extract rules from trained neural networks. In: Machine Learning Proceedings 1994, pp. 37–45. Elsevier (1994). https://doi.org/10.1016/B978-1-55860-335-6.50013-1
7. Craven, M.W., Shavlik, J.W.: Extracting tree-structured representations of trained networks. In: Touretzky, D.S., Mozer, M.C., Hasselmo, M.E. (eds.) Advances in Neural Information Processing Systems 8. Proceedings of the 1995 Conference, pp. 24–30. The MIT Press (Jun 1996), http://papers.nips.cc/paper/1152-extracting-tree-structured-representations-of-trained-networks.pdf
8. European Commission: AI Act – Proposal for a regulation of the european parliament and the council laying down harmonised rules on artificial intelligence (Artificial Intelligence Act) and amending certain union legislative acts. https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:52021PC0206 (2021)
9. European Commission, Directorate-General for Communications Networks, C., Technology: Ethics guidelines for trustworthy AI. Publications Office (2019). https://doi.org/doi/10.2759/346720
10. Franco, L., Subirats, J.L., Molina, I., Alba, E., Jerez, J.M.: Early breast cancer prognosis prediction and rule extraction using a new constructive neural network algorithm. In: Computational and Ambient Intelligence (IWANN 2007). LNCS, vol. 4507, pp. 1004–1011. Springer (2007). https://doi.org/0.1007/978-3-540-73007-1_121
11. Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., Pedreschi, D.: A survey of methods for explaining black box models. ACM Computing Surveys **51**(5), 1–42 (2018). https://doi.org/10.1145/3236009
12. Gunning, D., Aha, D.: Darpa's explainable artificial intelligence (xai) program. AI magazine **40**(2), 44–58 (2019)
13. Huysmans, J., Baesens, B., Vanthienen, J.: ITER: An algorithm for predictive regression rule extraction. In: Data Warehousing and Knowledge Discovery (DaWaK 2006). pp. 270–279. Springer (2006). https://doi.org/10.1007/11823728_26

14. Kenny, E.M., Ford, C., Quinn, M., Keane, M.T.: Explaining black-box classifiers using post-hoc explanations-by-example: The effect of explanations and error-rates in XAI user studies. Artificial Intelligence **294**, 103459 (2021). https://doi.org/10.1016/j.artint.2021.103459
15. Mökander, J., Morley, J., Taddeo, M., Floridi, L.: Ethics-based auditing of automated decision-making systems: Nature, scope, and limitations. Science and engineering ethics **27**(4), 1–30 (2021)
16. Rudin, C.: Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. Nature Machine Intelligence **1**(5), 206–215 (2019). https://doi.org/10.1038/s42256-019-0048-x
17. Sabbatini, F., Calegari, R.: Symbolic knowledge extraction from opaque machine learning predictors: GridREx & PEDRO. In: Kern-Isberner, G., Lakemeyer, G., Meyer, T. (eds.) Proceedings of the 19th International Conference on Principles of Knowledge Representation and Reasoning, KR 2022, Haifa, Israel. July 31 - August 5, 2022 (2022), https://proceedings.kr.org/2022/57/
18. Sabbatini, F., Ciatto, G., Calegari, R., Omicini, A.: On the design of PSyKE: A platform for symbolic knowledge extraction. In: Calegari, R., Ciatto, G., Denti, E., Omicini, A., Sartor, G. (eds.) WOA 2021 – 22nd Workshop "From Objects to Agents". CEUR Workshop Proceedings, vol. 2963, pp. 29–48. Sun SITE Central Europe, RWTH Aachen University (Oct 2021), 22nd Workshop "From Objects to Agents" (WOA 2021), Bologna, Italy, 1–3 Sep. 2021. Proceedings
19. Sabbatini, F., Ciatto, G., Calegari, R., Omicini, A.: Symbolic knowledge extraction from opaque ML predictors in PSyKE: Platform design & experiments. Intelligenza Artificiale **16**(1), 27–48 (2022). https://doi.org/10.3233/IA-210120, https://doi.org/10.3233/IA-210120
20. Sabbatini, F., Ciatto, G., Omicini, A.: GridEx: An algorithm for knowledge extraction from black-box regressors. In: Calvaresi, D., Najjar, A., Winikoff, M., Främling, K. (eds.) Explainable and Transparent AI and Multi-Agent Systems. Third International Workshop, EXTRAAMAS 2021, Virtual Event, May 3–7, 2021, Revised Selected Papers, LNCS, vol. 12688, pp. 18–38. Springer Nature, Basel, Switzerland (2021). https://doi.org/10.1007/978-3-030-82017-6_2
21. Sabbatini, F., Ciatto, G., Omicini, A.: Semantic web-based interoperability for intelligent agents with PSyKE. In: Calvaresi, D., Najjar, A., Winikoff, M., Främling, K. (eds.) Proceedings of the 4th International Workshop on EXplainable and TRAnsparent AI and Multi-Agent Systems, Lecture Notes in Computer Science, vol. 13283, chap. 8, pp. 124–142. Springer (2022). https://doi.org/10.1007/978-3-031-15565-9_8
22. Sabbatini, F., Grimani, C.: Symbolic knowledge extraction from opaque predictors applied to cosmic-ray data gathered with LISA Pathfinder. Aeronautics and Aerospace Open Access Journal **6**(3), 90–95 (2022). https://doi.org/10.15406/aaoaj.2022.06.00145, https://doi.org/10.15406/aaoaj.2022.06.00145