Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 200 (2022) 216–226

Procedia Computer Science

www.elsevier.com/locate/procedia

3rd International Conference on Industry 4.0 and Smart Manufacturing

Explainable AI for Industry 4.0: Semantic Representation of Deep Learning Models

Vagan Terziyan [a]*, Oleksandra Vitko [b]

[a]*Faculty of Information Technology, University of Jyväskylä, 40014, Jyväskylä, Finland*

[b]*Depatment of Artificial Intelligence, Kharkiv National University of Radio Electronics, 61166, Kharkiv, Ukraine*

Abstract

Artificial Intelligence is an important asset of Industry 4.0. Current discoveries within machine learning and particularly in deep learning enable qualitative change within the industrial processes, applications, systems and products. However, there is an important challenge related to explainability of (and, therefore, trust to) the decisions made by the deep learning models (aka black-boxes) and their poor capacity for being integrated with each other. Explainable artificial intelligence is needed instead but without loss of effectiveness of the deep learning models. In this paper we present the transformation technique between black-box models and explainable (as well as interoperable) classifiers on the basis of semantic rules via automatic recreation of the training datasets and retraining the decision trees (explainable models) in between. Our transformation technique results to explainable rule-based classifiers with good performance and efficient training process due to embedded incremental ignorance discovery and adversarial samples ("corner cases") generation algorithms. We have also shown the use-case scenario for such explainable and interoperable classifiers, which is collaborative condition monitoring, diagnostics and predictive maintenance of distributed (and isolated) smart industrial assets while preserving data and knowledge privacy of the users.

Keywords: Explainable Artificial Intelligence; Industry 4.0; semantic web; deep learning; predictive maintenance

1. Introduction and Related Work

We all observe the huge impact made recently by the Artificial Intelligence (AI) while addressing the decision-making problems within the big data challenge context. People start to think whether their further interaction and integration with the AI goes towards the replacement rather than augmentation of human decision makers [1]. However, in spite of fast development of artificial, computational, autonomous and smart decision-making

* Corresponding author. Tel.: +358-50-373-2127; E-mail address: vagan.terziyan@jyu.fi

10.1016/j.procs.2022.01.220

capabilities, models and tools, the decision-making will remain human-centric even during the transformation from Industry 4.0 to Industry 5.0 [2]. According to [3], a new revolutionary wave—Industry 5.0 will be an "Age of Augmentation" or the human-machine symbiosis.

Almost all the processes within modern industry, including smart manufacturing, Industry 4.0 and 5.0, are driven by the decision-making where the balance of the decision power must be smartly distributed among the people involved, AI and Machine Learning (ML) models, autonomous agents, robots, and other smart components, however, preserving the key decisive role of a human. Bruzonne et al. [4] suggested strategic engineering as a discipline to study synergy between the Industry 4.0 and various decision-making paradigms where AI will play the key role together with modelling, simulation and data analytics. Industry 4.0 requires properly designed AI models to support humans to handle various decision-making problems, from the everyday manufacturing decisions to the strategic ones and from the business as usual decisions to the decisions in critical emergencies [5], from individual decision models to collaborative and resilient decision infrastructures to handle crises, such as COVID-19 pandemics [6]. According to Longo et al. [7], various human factors are still very important in Industry 4.0, even within highly automated processes, such as intelligent fault detection and alarm management. Popular and important problem of predictive maintenance also requires human involvement because the real-world implementations are still rare [8] due to lack of high-quality monitoring data and poor user experience in applying AI methods and tools.

Therefore, a human remains to be a key player in the industrial decision-making processes, both as component within a heterogeneous decision-making "team" and as a user of the decisions made. However, this means that the human, to be able to collaborate during decision-making and to apply the decision afterwards, must be well aware of not only what the decision is but also how it was made and why. As noticed in [9], humans who are using the real-time process predictions within smart factories need to establish confidence in the automatically derived predictions and suggested decisions. Explainable AI (XAI) has emerged to enable humans to understand, trust, and manage the AI they work with. Therefore, there is a great demand of explicit declarative knowledge linked to the large ontologies [10] to enable transparency, accountability [11] and trust for the industrial AI/ML applications [12]. However, as noticed in [13], the major explainability problems are related to the deep learning models used within the cyber-physical systems context and based on artificial neural networks where the relational link between input and output is not observable.

In contrast to (deep) neural networks, decision trees [14] are known to be a type of explainable models produces by the ML processes. Decision trees combine simple questions about the data in an understandable way and, therefore, their formal representation enables automatic extraction of the explicit decision rules from them. John Ross Quinlan, in addition to the famous C4.5 algorithm [15] widely used to learn decision trees from data, suggested also an approach [16] for transforming such trees into small sets of production rules (suitable and transparent formalism for expressing knowledge in industrial diagnostics and maintenance systems). However, his approach requires the training set of cases from which the decision tree has been generated. Direct model-to-model transformation between the decision tree and the rules' set is more challenging but doable task [17].

However, it is also important that the rules coming from different (trained independently) decision trees could be seamlessly and incrementally integrated with other (previously available) rules. Such integration requires sharing common ontologies among rule sets and representing the rules in a standardized form such as, e.g., Semantic Web Rule Language (SWRL) [18]. Just because of little interoperability between former rule-based systems, the SWRL has been designed as a language for sharing rules [19]. SWRL rules can be updated incrementally because SWRL language supports the Open World Assumption and monotonic reasoning, and has already a good reputation among industry experts. For example, [20] reports on developed domain ontology for modeling predictive maintenance knowledge and a set of SWRL predictive rules for reasoning about the time and criticality of machinery failures. In [21], the attempt is reported aiming to capture SWRL rules directly from the decision tree trained by WEKA ML-as-a-service for the problem of managing order allocation in manufacturing network containing 300 small enterprises. The importance of applying semantic technologies within Industry 4.0 has been argued in [22], especially regarding the automation of manufacturing processes and intelligent condition monitoring systems on top of them. Also [23] argues that combination of ML models and semantic web technologies (ontologies and reasoning) is important for predictive maintenance problems. Both of these problems (condition monitoring together with predictive maintenance) require models' integration because these models are usually learned independently from data coming from different items of the same asset class, e.g., electric motors of same or similar type [24]. Another important area within Industry 4.0, where the explainable and easy-to-integrate models are needed, is related to robots and autonomous systems in general and agents in particular, and which requires interoperable simulations focusing on predictive maintenance and security important for autonomous critical missions [25]. Generic integration and interoperability solution is reported in [26], where semantic reasoning is performed by autonomous agents used as a proxy to smart industrial assets with embedded intelligence [27]. Good summary on the current state of the semantic web research and applications (after 20 years of it history) is available in [28]; check also the cross-domain multidisciplinary reviews on the benefits of semantics for cyber-physical systems in general [29] and for cyber-physical production systems in particular [30]. All the reviews indicate the great need for semantic technologies in data and knowledge representation and reasoning applied to modern industry due to highest level of explainability and integration capacity of these technologies.

Other option of integrating and using collaboratively several distinct ML models trained from isolated decentralized (due to the ownership or privacy reasons [31]) datasets is related to the so-called federated learning [32]. When the component models themselves are not explainable, then the explainability of the decisions produced by federated learning processes is a complex task and an additional challenge for XAI [33].

Therefore, the main challenge on the roadmap towards XAI remains the transformation of ML models (individual or collaborating) between the connectionist (e.g., deep neural networks) and symbolic (e.g., decision trees) representations. While there exist several solutions for transformation of decision trees to neural networks (see [34]) and also efficient hybrids of decision trees and neural networks (see [35] or [36]), however, the task of making the neural networks explainable by transforming them into the decision trees is still on the table. Symbolic models, which are based on formal logic and deductive reasoning are fundamentally different from the models based on neural networks. As noticed in [37], this difference is not only in their approach and internals, but also in their capabilities. Therefore, neural-symbolic integration aims to bridge and benefit from both paradigms. Artificial neural networks are the black-boxes lacking interpretability however they are known to generalize better than the decision trees. Some attempts have been reported regarding approaching the rule extraction from neural nets (see, e.g., [38], [39], [40] and [41]), however, the problem still remains largely unresolved and unexplored.

In this paper, we formulate our generic objectives with the following assumptions. Assume that we have some complex decision-making problem within some of the Industry 4.0 domains. Assume that various evidence, former observations (data) has been collected by independent business players and used for training various heterogeneous ML models, majority of which are not explainable (aka “black-boxes”). Assume that all these individual models become available as services for making further decisions within the domain, however all the data they were trained from and the models’ internals are not available. The objectives are: (a) to capture/extract the hidden individual models as XAI (decision trees); (b) make automatic transformation “decision trees – SWRL rules” under umbrella of the domain ontology; (c) integrate the SWRL models for collaborative decision making. We address these objectives in the paper and provide the use case scenario – industrial assets’ condition monitoring, diagnostics and predictive maintenance, in which such objectives make sense and could work for similar cases within Industry 4.0.

This article is structured as follows: Section 2 describes the basic intuition behind our approach; Section 3 offers the way to represent the decision trees as SWRL rules under umbrella of some domain ontology; Section 4 provides the basic schema of transformation between the black-box models and XAI models preserving the classification accuracy and gives an example of the use case scenario (predictive maintenance); and we conclude in Section 5.

2. Intuition behind the Approach

The intuition behind the approach to our objectives is explained with an example in Figure 1. Assume that we have 20 training samples as data available for supervised ML in two-dimensional decision space (attributes X and Y), from which 10 samples are of class “Black” and 10 of class “Grey” as shown in Figure 1(a). If to train some deep feedforward neural network with such data we will get the decision boundary, which smoothly separates Grey and Black samples as shown in Figure 1(b). It is known that neural networks usually generalize well, i.e., they are capable of finding the balance between the overfitting and underfitting while training and, therefore, perform better than other models with the test sets. We used the TensorFlow open-source code in our experiments for training neural networks. Let us now forget for a moment about the neural network and try to train independently a simple decision tree on the basis of the same 20 training samples and represent it (branch-by-branch) in the form of rules. In our experiments, we used the basic decision tree learning algorithms for batch training process [42] available online via [43] and also the algorithms from [44] when simulating streaming data.

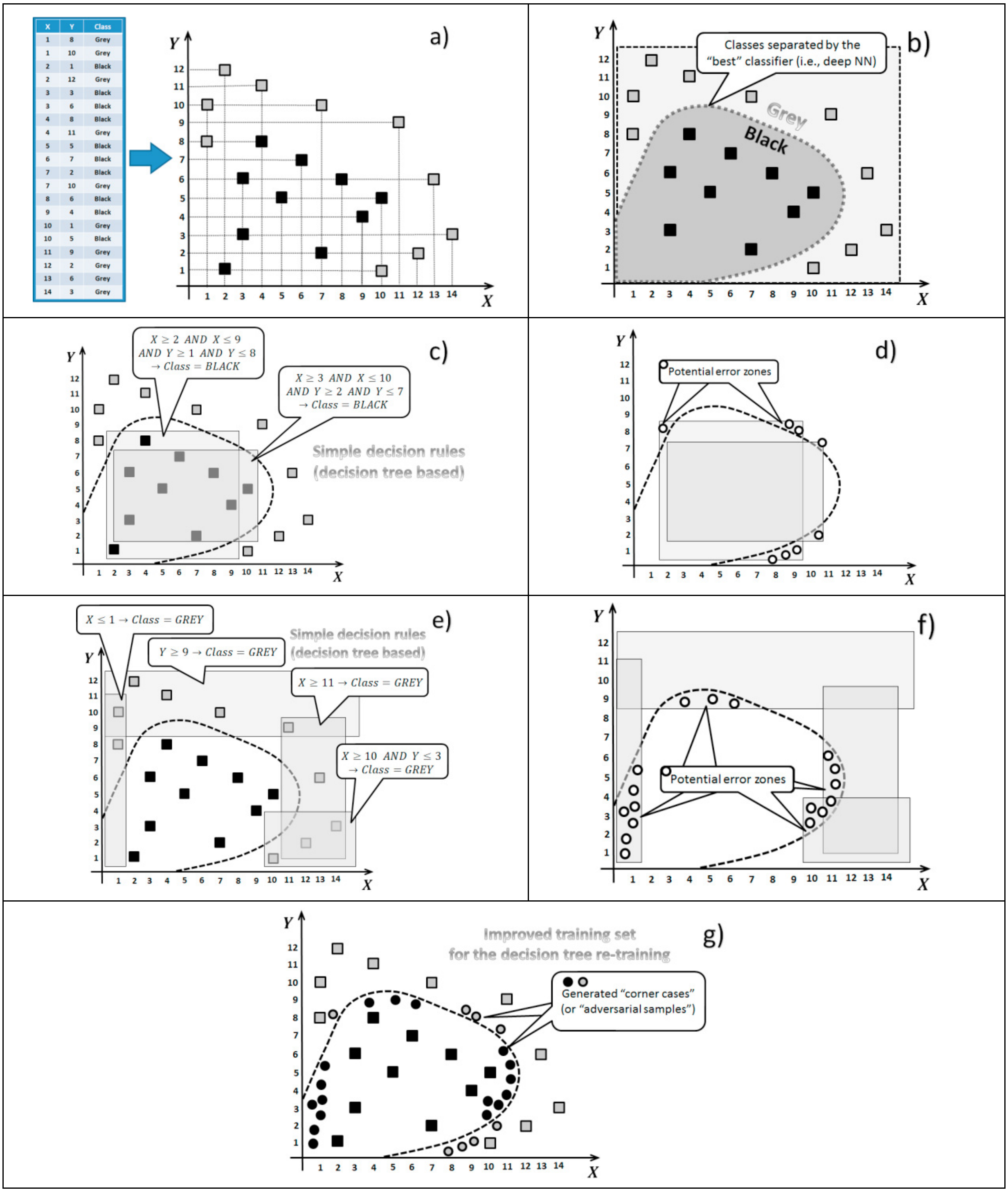

Figure 1. Neural network vs. decision tree example: (a) 20 training samples; (b) decision boundary produced by neural network; (c) two rules for the class "Black" produced by the trained decision tree; (e) potential error zones for the decision tree (regarding class "Black") where the trained neural network performs better; (e) two rules for the class "Grey" produced by the trained decision tree;.(f) potential error zones for the decision tree (regarding class "Grey") where the trained neural network performs better; (g) generated samples ("corner cases" or "adversarial samples") within the discovered potential error zones that could be used to re-train the decision tree aiming better classification accuracy and robustness.

For the example in Figure 1, the trained decision tree contains six rules: two for the class “Black” as shown in Figure 1(c); and four rules for the class “Grey” as shown in Figure 1(e). As one can see, each rule corresponds to an appropriate rectangle in two-dimensional space containing the training samples with the corresponding class labels. If to assume that the trained neural network is an ideal classifier for our example, then one can notice that the trained decision tree model underfits a bit the available data. There are potential error zones (mismatch areas between the decision boundaries drawn by the “ideal” neural network and by the decision tree) within the decision space, samples from which (if given as the test samples) will be misclassified by the decision tree. Figure 1(d) shows such error zones (and potentially misclassified test samples drawn as circles) for the rules related to the class “Black”, and Figure 1(f) shows error zones for the rules associated with the class “Grey”. Finally, we may assume that, if to enhance the training set for the decision tree learning by adding correctly labelled samples (often called as “corner cases” or “adversarial samples” as shown in Figure 1(g)) from these potential error zones and re-train the decision tree model accordingly, then the decision tree classifier will be much more precise (having the robust decision boundary closer to the “ideal” one from the neural network).

The observation above suggests the following approach of representing a black-box (e.g., deep neural network) ML model into an explainable decision tree (i.e., potentially, into a set of rules). We have to: “interview” the black-box (available as a service) with “smart” (targeting potential error zones) queries; get class labels from the black-box as an outcome and make the corner cases’ samples; create the training set from these samples for the decision tree learning; and, finally, train the decision tree. Therefore, we can use the black-box classifier as an “expert” to label the samples needed for the training dataset for supervised learning of a decision tree. The key point here is that we are not asking our black-box just some arbitrary questions and by doing so making the “interview” process long and “blind”. On the contrary, we use smart way to discover in real time the “confusion” zones during the interview, which are located along the decision boundary between classes or along the boundary between the desired decision space and “the rest of the world” and ask targeted queries aiming to get labelled corner cases (or adversarial samples). This way not only enables fast creation of training data (efficiency) for the decision tree learning but also guarantees effectiveness and robustness of the trained decision tree classifiers. The suggested schema and algorithm for the black-box—XAI transformation will be presented in Section 4.

3. From the Decision Trees to SWRL Rules

We experimented with the SWRL representations of the decision trees within Protégé [45], which is a popular open-source environment for designing ontologies and maintaining ontology-based systems. To extend capabilities of Protégé, we developed the XAI-plugin responsible for generating SWRL rules from data via decision tree learning. The plugin utilizes C4.5 ML algorithm as a remote service, trains decision tree from data and represents decision tree in the form of SWRL rules so that classes and properties from these rules refer to particular classes and properties already defined in the domain ontology designed in Protégé. Assume that the taxonomy of classes in the domain ontology includes class C, which is defined as a disjoint union of subclasses $C_1, C_1, \dots, C_n$. Assume that there are some instances with their datatype properties (attribute-value pairs) registered in each of the subclasses. The plugin organizes all these instances with their datatype properties and labels of the subclasses as a training set and automatically calls the remote C4.5 service to train the decision tree. Then the outcome from the remote service is represented in the form of SWRL rules using special procedure. The resulting rule set will later work as an embedded classifier within the ontology. When certain instance with its properties appears in class C, then this classifier (SWRL rule set) will automatically concretize it and re-register it to one of the subclasses $C_1, C_1, \dots, C_n$.

Figure 2 shows our previous example with some Protégé screenshots to demonstrate the SWRL rules automatically generated by the XAI-plugin. These rules are capable to classify any arbitrary instance as “Black” or “Grey” given the values of its properties (X and Y). As we discussed already, the quality of the SWRL (decision tree-based) classifier will be much higher if the set of training samples would include more adversarial samples closer to the decision boundary between classes. Therefore, to fill the subclasses $C_1, C_1, \dots, C_n$ with potential (challenging) training instances we use our approach of “interviewing” a strong black-box classifier (e.g., deep neural network) by generating and labelling the adversarial samples and feeding the ontology with such samples to enable XAI-plugin to design stronger SWRL classifiers. This will be shown in the following section.

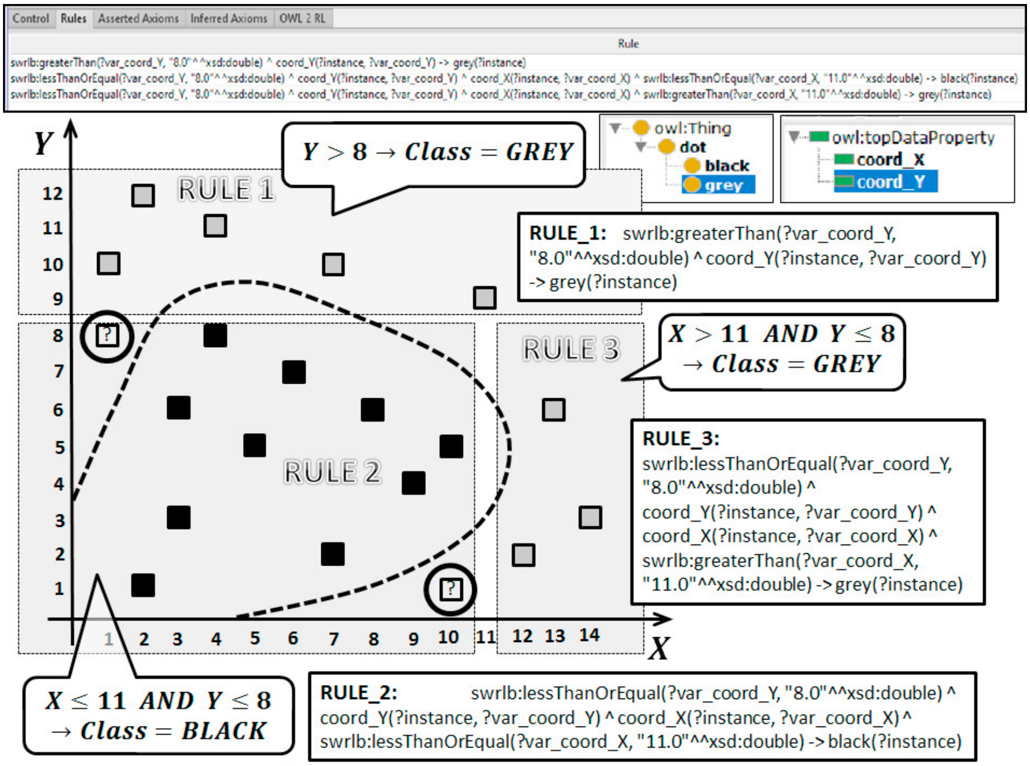

Figure 2. SWRL rules generated on the basis of 20 training samples using decision tree learning algorithm and special Protégé plugin

4. "Black-Box" – XAI Transformation Schema and the Use-Case Scenario

Complete schema for intended transformation (i.e., "cloning" the black-box ML models into semantic XAI models) is shown in Figure 3. First, the shape of the multidimensional decision space is defined, which is the domain boundary separating the potential data manifolds representing intended classes with the "rest of the world". With such boundary we specify the area, in which we are going to train/test/use our intended classifier. Usual shapes are rectangular (in general case - hyperrectangle), circular (in general case - hypersphere) or (seldom) arbitrary (analytically defined multidimensional closed curved hyperplane).

The next key component of the schema is "Ignorance Explorer". Its role is to discover in real time the maximal void (ignorance/confusion zone) within the decision space and report coordinates of the focus of that zone as an adversarial sample to be labelled. For the ignorance discovery we used algorithm provided in [46]. The analytics from [46] is defined so that it works with an arbitrary shape of the domain boundary. The intended ignorance zones (where the potential adversarial samples are located) exist and must be discovered not only between the differently labelled data manifolds (i.e., groups of samples with different class labels) but also between these manifolds and the decision space boundary. Therefore, the ignorance discovery analytics from [46] considers two types of voids within the decision space: (1) empty hyperspheres centered on the domain boundary and touching at least one data sample; and (2) empty hyperspheres placed completely within the decision space and touching two differently labelled data samples. The algorithm for the ignorance focus (adversarial sample) discovery at each iteration of the "black-box interview" works as follows. At every iteration step, the ignorance zones (voids) are discovered within the domain populated with the already nominated adversarial samples (starting from the empty set of samples and adding one at each iteration). The domain boundary could be set up manually by the data scientist or it can be discovered as the (minimal volume) multidimensional hypersphere or hyperrectangle, which covers all the data samples. The center of the largest void (aka adversarial sample) is discovered and its coordinates are used as a query (submitted by the

“Challenger” component in Figure 3) to the black-box classifier, which returns label to the discovered adversarial sample. Labelled sample is added to the set of already nominated adversarial samples and the process of ignorance discovery will be repeated with the updated set. When we reach the stopping criterion (i.e., the radius of the largest available void becomes less than the predefined minimum) and, therefore, the intended number of adversarial samples is collected as a training set for supervised learning, then we can continue with generating the decision tree and corresponding SWRL rules as described in the previous section. Finally, as shown in Figure 3, we will get the explainable SWRL “clone” of the original black-box (deep neural network) classifier, which will provide good classification accuracy in addition to explainability and, as an added value will have the capacity to be integrated with similar classifiers.

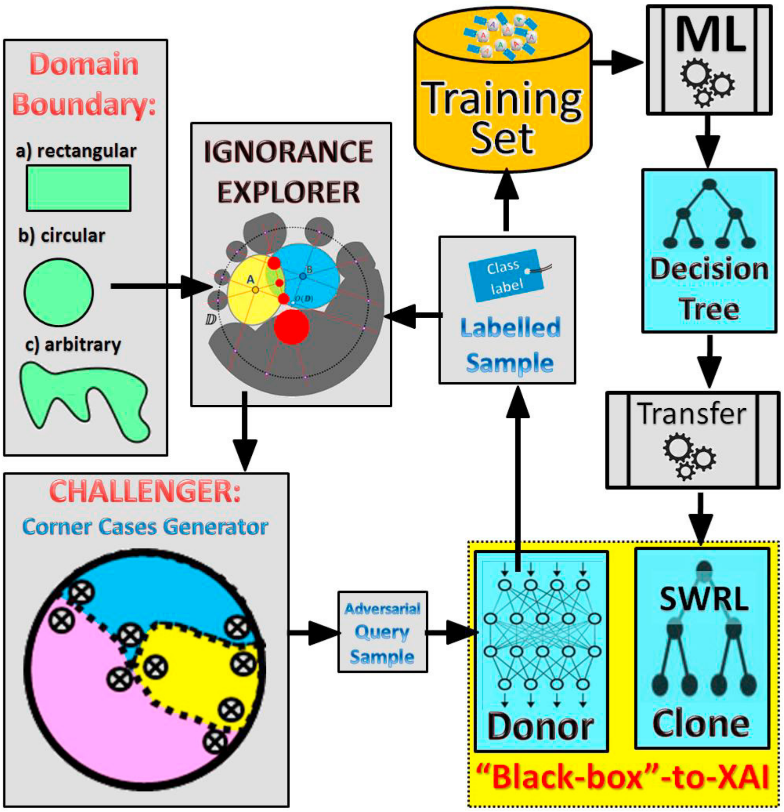

Figure 3. The generic schema of “cloning” black-box classification models to the explainable form of SWRL rules

Possible use-case scenario where one can get benefits of using the schema above for Industry 4.0 is the condition monitoring, diagnostics and maintenance of industrial assets as shown in Figure 4. Here is the case when the smart and expensive assets of the same type have been sold and used globally. The automatic diagnostic models are continuously trained independently by local data scientists and used for predictive maintenance. These models are naturally heterogeneous black-boxes and their users are not going to share training data with each other. However, it would be beneficial for all to use (for more advanced diagnostics) some integrated model composed of these individual diagnostic models. This can be done by making the XAI-SWRL clones of the available black-boxes and use the integration capacity of SWRL to generate an integrated (more capable) model for diagnostics and predictive maintenance of the individual instances of the industrial asset (Figure 4). We have implemented such schema as a new additional feature of our UBIWARE platform [47], which is a semantic middleware for the internet of things. Possible conflicts between individual rules are resolved using the dynamic integration technique [48] when the most reputed (within a particular subspace of the decision space) rule is applied to handle particular classification case. Reputation (an expected performance within certain subspace) of each rule is computed by special procedure [48].

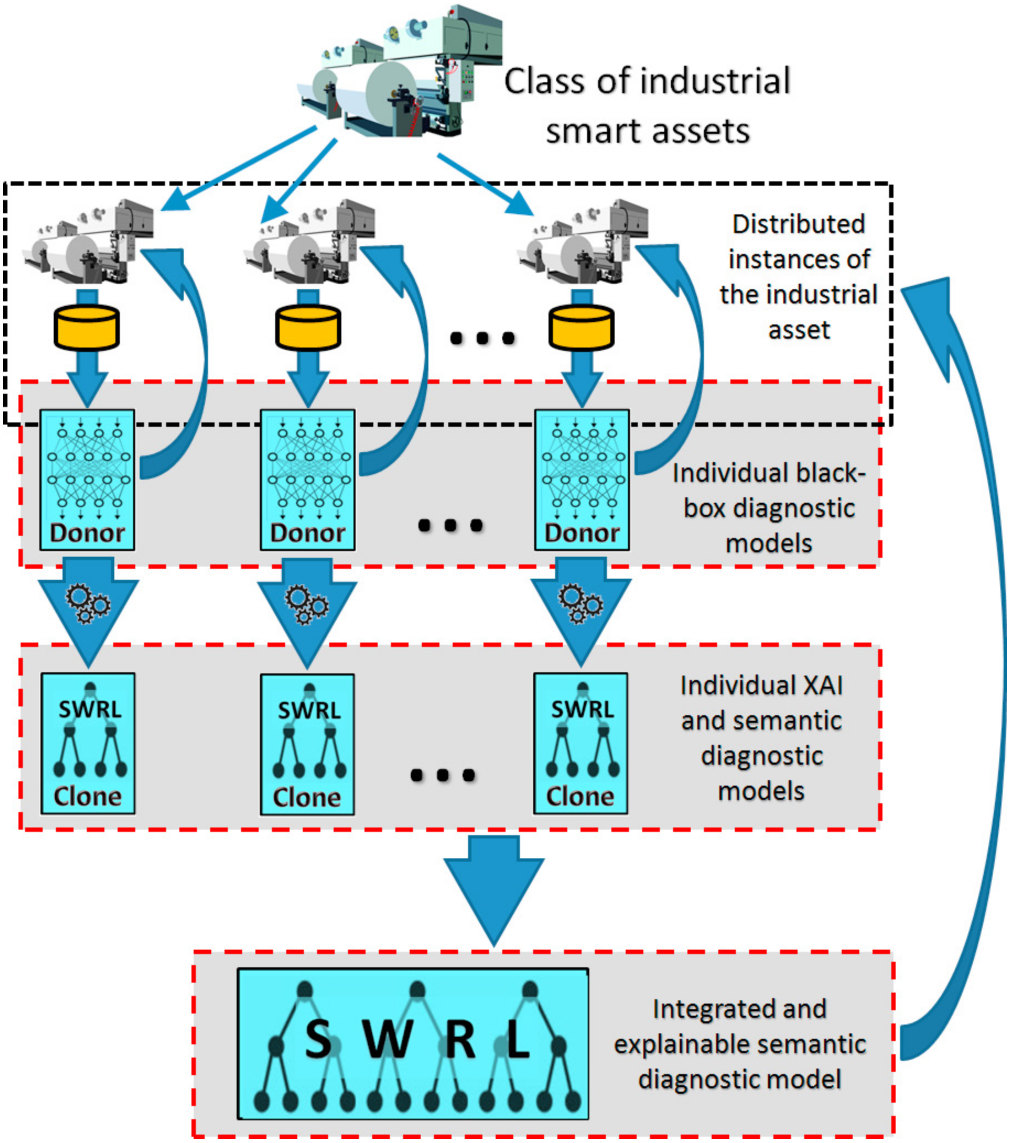

Figure 4. Use case scenario (predictive maintenance of smart industrial assets) for the black-box – XAI (SWRL) transformation and integration

In this study, we took the real industrial use-cases from our former projects SmartResource and UBIWARE (complete documentation and downloads can be found in: www.cs.jyu.fi/ai/SmartResource_UBIWARE.html). During these projects' times, we have collected huge datasets on industrial assets (paper machines, power networks, etc.). Due to the explainability requirement, originally, we have used either manually-fed expert rules for fault detection and classification or ML algorithms (Bayesian networks and decision tree learning) [49]. With such models, that time, we could get the overall classification accuracy of about 75-85%. In the current study we reuse the same datasets, however we were able to train the deep neural networks, which gave us much higher classification accuracy (85-97%). Then we used our XAI transformation technique and get the explainable version of these deep neural networks in terms of SWRL rules. By this way we were able to preserve required explainability but in the same time we were able to use the advantage of neural network (i.e., capability to generalize better and provide better classification accuracy).

5. Conclusions

In this paper, we have used several AI and ML techniques from our former research to approach the problem of explainability (XAI), which is important nowadays for smart manufacturing and Industry 4.0. We have shown that outcomes of deep learning, which are usually very capable and accurate but not explainable (aka black-boxes) AI models, can be automatically retrained as decision trees without access to the original training data and without essential loss of generalization accuracy. Training data for potential decision trees must be recovered by "interviewing" these black-boxes. To make the process of training data recovery efficient and for improving the quality of training data, we applied special incremental ignorance discovery algorithms from [46] where the ignorance or confusion zones (specially defined "voids") within the incrementally designed training dataset are discovered as a target area to generate new queries to the black-box at each iteration aiming to label most vulnerable places for potential decision tree classifier.

We have extended our UBIWARE middleware for the IoT [47] with the appropriate implementation. We have shown that, for better explainability and integration capacity, the retrained decision tree models could be represented as rules and more specifically as SWRL rules, which support the Open World Assumption, could be connected with the domain ontology and other previously defined rules, i.e., enabling semantic integration and interoperability of AI models and knowledge-based systems.

We have made the transformation algorithm: neural network – SWRL rules with the explicit mediator - a decision tree aiming to get XAI representation in both forms (decision tree and SWRL) each potentially useful in different applications.

As a use-case scenario, we considered the popular problem of industrial assets' condition monitoring, diagnostics and predictive maintenance where not only explainability of ML-driven tools and models is an issue but also the distributed ownership of various and heterogeneous diagnostic models. We have shown that "cloning" each individual model into SWRL rules' representation and linking these rules together results to more advanced collaborative diagnostics (aka special kind of federated learning technique) of industrial assets beneficial for all distributed users without loss of their privacy and ownerships.

In this paper we considered only models originated from the numeric data, which is current limitation of the approach. The presented procedure can be used with the numeric data, e.g., an input vector for an artificial (feed-forward) neural network or the data where each attribute is related to some feature of the object being analyzed and with the concrete numeric value for the feature. Number of dimensions could be arbitrary (in our examples we used 2D data just to ease readers' perceptions of the figures). In our use cases we have experimented with data containing 100-150 attributes. However, regarding other data formats, e.g., the image data, where initial information about the object comes from the values of the pixels and which requires convolutional layers of a neural network to discover and get the values for hidden actual features, our procedure cannot be used as such yet and will need further modifications. In near future we are going to adapt the approach for categorical data, images, video, text and speech.

We limited this study with the classification problems only and leave the regression tasks for the future due the following couple of reasons: (a) We used a decision tree as a mediator between the neural networks and the SWRL rules. Decision tree is a type of supervised learning algorithm, which has a pre-defined target variable, and, therefore, such trees could be used for regression problems if and only if the target variable is inside the range of values seen in the dataset used for training; (b) Current implementation of Pellet (reasoner for SWRL) has certain difficulties while re-assigning numeric values for the datatype attributes (needed for regression) due to Open World assumption and, therefore, limitations for the non-monotonic inferences.

The limitations of our reported algorithm (in quantitative terms) are mainly related to the resource-consuming "Ignorance Explorer" component of data processing. For the dataset containing n data samples with d attributes (d-dimensional data space), the computational complexity of the maximal void (ignorance/confusion zone) discovery could be estimated as $O(n^3\ d)$ as has been shown in [46].

References

[1] Duan Y., Edwards J. S., & Dwivedi Y. K. (2019). "Artificial intelligence for decision making in the era of Big Data–evolution, challenges and research agenda". *International Journal of Information Management*, **48**, 63-71. doi:10.1016/j.ijinfomgt.2019.01.021

[2] Nahavandi S. (2019). "Industry 5.0—A human-centric solution". *Sustainability*, **11(16)**, 4371. doi:10.3390/su11164371

[3] Longo F., Padovano A., & Umbrello S. (2020). "Value-Oriented and Ethical Technology Engineering in Industry 5.0: A Human-Centric Perspective for the Design of the Factory of the Future". *Applied Sciences*, **10(12)**, 4182. doi:10.3390/app10124182

[4] Bruzzone A., Massei M., & Sinelshnkov K. (2020). "Enabling strategic decisions for the industry of tomorrow". *Procedia Manufacturing*, **42**, 548-553. doi:10.1016/j.promfg.2020.02.028

[5] Longo F., Nicoletti L., & Padovano A. (2019). "Emergency preparedness in industrial plants: A forward-looking solution based on industry 4.0 enabling technologies". *Computers in Industry*, **105**, 99-122. doi:10.1016/j.compind.2018.12.003

[6] Shen W., Yang C., & Gao L. (2020). "Address business crisis caused by COVID-19 with collaborative intelligent manufacturing technologies". *IET Collaborative Intelligent Manufacturing*, **2(2)**, 96-99. doi:10.1049/iet-cim.2020.0041

[7] Longo F., Padovano A., Gazzaneo L., Frangella J., & Diaz R. (2021). "Human factors, ergonomics and Industry 4.0 in the Oil&Gas industry: a bibliometric analysis". *Procedia Computer Science*, **180**, 1049-1058. doi:10.1016/j.procs.2021.01.350

[8] Zenisek J., Holzinger F., & Affenzeller M. (2019). "Machine learning based concept drift detection for predictive maintenance". *Computers & Industrial Engineering*, **137**, 106031. doi:10.1016/j.cie.2019.106031

[9] Rehse J. R., Mehdiyev N., & Fettke P. (2019). "Towards Explainable Process Predictions for Industry 4.0 in the DFKI-Smart-Lego-Factory". *KI-Künstliche Intelligenz*, **33(2)**, 181-187. doi:10.1007/s13218-019-00586-1

[10] Holzinger A. (2018). "From Machine Learning to Explainable AI". In: *Proceedings of the 1st World Symposium on Digital Intelligence for Systems and Machines* (pp. 55-66). IEEE. doi:10.1109/DISA.2018.8490530

[11] Shin D. (2021). "The Effects of Explainability and Causability on Perception, Trust, and Acceptance: Implications for explainable AI". *International Journal of Human-Computer Studies*, **146**, 102551. doi:10.1016/j.ijhcs.2020.102551

[12] Gade K., Geyik S. C., Kenthapadi K., Mithal V. & Taly A. (2019). "Explainable AI in Industry". In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (pp. 3203-3204). ACM. doi:10.1145/3292500.3332281

[13] Daglarli E. (2021). "Explainable Artificial Intelligence (xAI) Approaches and Deep Meta-Learning Models for Cyber-Physical Systems". In: *Artificial Intelligence Paradigms for Smart Cyber-Physical Systems* (pp. 42-67). IGI Global. doi:10.4018/978-1-7998-5101-1.ch003

[14] Kingsford C., Salzberg S. L. (2008). "What are Decision Trees?". *Nature Biotechnology*, **26(9)**, 1011-1013. doi:10.1038/nbt0908-1011

[15] Quinlan J.R. (1992). "*C4.5 Programs for Machine Learning*". San Mateo, CA: Morgan Kaufmann.

[16] Quinlan J. R. (1987). "Generating Production Rules from Decision Trees". In: *Proceedings of the 10th International Joint Conferences on Artificial Intelligence* (Vol. 87, pp. 304-307). doi:10.5555/1625015.1625078

[17] Mitchell T. M. (1997). "*Machine Learning*". McGraw-Hill.

[18] Horrocks I., Patel-Schneider P. F., Boley H., Tabet S., Grosof B., & Dean M. (2004). "SWRL: A semantic web rule language combining OWL and RuleML". *W3C Member submission*, **21(79)**, 1-31.

[19] O'Connor M., Knublauch H., Tu S., Grosof B., Dean M., Grosso W., & Musen M. (2005). "Supporting rule system interoperability on the semantic web with SWRL". In: *Proceedings of the International Semantic Web Conference* (pp. 974-986). Springer, Berlin, Heidelberg. doi:10.1007/11574620_69

[20] Cao Q., Samet A., Zanni-Merk C., de Beuvron F. D. B., & Reich C. (2019). "An Ontology-Based Approach for Failure Classification in Predictive Maintenance using Fuzzy C-means and SWRL Rules". *Procedia Computer Science*, **159**, 630-639. doi:10.1016/j.procs.2019.09.218

[21] Khan Z. M. A., Saeidlou S., & Saadat M. (2019). "Ontology-based decision tree model for prediction in a manufacturing network". *Production & Manufacturing Research*, **7(1)**, 335-349. doi:10.1080/21693277.2019.1621228

[22] Cao Q., Giustozzi F., Zanni-Merk C., de Bertrand de Beuvron F., & Reich C. (2019). "Smart Condition Monitoring for Industry 4.0 Manufacturing Processes: An Ontology-Based Approach". *Cybernetics and Systems*, **50(2)**, 82-96. doi:10.1080/01969722.2019.1565118

[23] Dalzochio J., Kunst R., Pignaton E., Binotto A., Sanyal S., Favilla J., & Barbosa J. (2020). "Machine Learning and Reasoning for Predictive Maintenance in Industry 4.0: Current Status and Challenges". *Computers in Industry*, **123**, 103298. doi:10.1016/j.compind.2020.103298

[24] Paolanti M., Romeo L., Felicetti A., Mancini A., Frontoni E., & Loncarski J. (2018). "Machine Learning Approach for Predictive Maintenance in Industry 4.0". In: *Proceedings of the 14th IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications* (pp. 1-6). IEEE. doi:10.1109/MESA.2018.8449150

[25] Bruzzone A. G., Longo F., Agresta M., Di Matteo R., & Maglione G. L. (2016). "Autonomous systems for operations in critical environments". In: *Proceedings of the Modeling and Simulation of Complexity in Intelligent, Adaptive and Autonomous Systems 2016 (MSCIAAS 2016) and Space Simulation for Planetary Space Exploration (SPACE 2016)* (pp. 1-8). Society for Computer Simulation International. doi:10.5555/2962664.2962667

[26] Terziyan V., & Zharko A. (2003). "Semantic Web and Peer-To-Peer: Integration and Interoperability in Industry". *International Journal of Computers, Systems and Signals*, **4**(**2**), 33-46. https://www.mit.jyu.fi/agora-center/inbct/InBCT31/IJCSS-2003.pdf

[27] Terziyan V. (2008). "SmartResource–Proactive Self-Maintained Resources in Semantic Web: Lessons Learned". *International Journal of Smart Home*, **2**(**2**), 33-57. https://gvpress.com/journals/IJSH/vol2_no2/3.pdf

[28] Hitzler P. (2021). "A review of the semantic web field". *Communications of the ACM,* **64**(**2**), 76-83. doi:10.1145/3397512

[29] Sabou M., Biffl S., Einfalt A., Krammer L., Kastner W., & Ekaputra F. J. (2020). "Semantics for Cyber-Physical Systems: A cross-domain perspective". *Semantic Web*, **11**(**1**), 115-124. doi:10.3233/SW-190381

[30] Sabou M., Ekaputra F. J., & Biffl S. (2017). "Semantic web technologies for data integration in multi-disciplinary engineering". In: *Multi-disciplinary engineering for cyber-physical production systems* (pp. 301-329). Springer, Cham. doi:10.1007/978-3-319-56345-9_12

[31] Yang Q., Liu Y., Chen T., Tong Y. (2019). "Federated Machine Learning: Concept and Applications". *ACM Transactions on Intelligent Systems and Technology*, **10**(**2**), 1-19. doi:10.1145/3298981

[32] Zhang C., Xie Y., Bai H., Yu B., Li W., Gao Y. (2021). "A Survey on Federated Learning". *Knowledge-Based Systems*, **216**, 106775 doi:10.1016/j.knosys.2021.106775

[33] Rieke N., Hancox J., Li W., Milletari F., Roth H. R., Albarqouni S., ... & Cardoso M. J. (2020). "The future of digital health with federated learning". *NPJ Digital Medicine*, **3**(**1**), 1-7. doi:10.1038/s41746-020-00323-1

[34] Setiono R., & Leow W. K. (1999). "On Mapping Decision Trees and Neural Networks". *Knowledge-Based Systems*, **12**(**3**), 95-99. doi:10.1016/S0950-7051(99)00009-X

[35] Zorman M., & Kokol P. (2002). "Hybrid NN-DT Cascade Method for Generating Decision trees from Backpropagation Neural Networks". In: *Proceedings of the 9th International Conference on Neural Information Processing* (vol. 4, pp. 2003-2007). IEEE. doi:10.1109/ICONIP.2002.1199024

[36] Wan A., Dunlap L., Ho D., Yin J., Lee S., Jin H., Petryk S., Bargal S. A., & Gonzalez J. E. (2020). "NBDT: neural-backed decision trees". *arXiv preprint arXiv:2004.00221*. https://arxiv.org/pdf/2004.00221.pdf

[37] Hitzler P., Bianchi F., Ebrahimi M., & Sarker M. K. (2020). "Neural-symbolic integration and the semantic web". *Semantic Web,* **11**(**1**), 3-11. doi:10.3233/SW-190368

[38] Taha I. A. & Ghosh J. (1999). "Symbolic interpretation of artificial neural networks". *IEEE Transactions on Knowledge and Data Engineering*, **11** (**3**), 448–463. doi:10.1109/69.774103

[39] Gupta A., Park S., & Lam S. M. (1999). "Generalized analytic rule extraction for feedforward neural networks". *IEEE Transactions on Knowledge and Data Engineering*, **11** (**6**), 985–991. doi:10.1109/69.824621

[40] Hruschka E. R., & Ebecken N. F. (2006). "Extracting rules from multilayer perceptrons in classification problems: A clustering-based approach". *Neurocomputing*, **70** (**1-3**). 384–397. doi:10.1016/j.neucom.2005.12.127

[41] Nguyen T. D., Kasmarik K. E., Abbass H. A. (2021). "Towards Interpretable Deep Neural Networks: An Exact Transformation to Multi-Class Multivariate Decision Trees". *arXiv preprint* arXiv:2003.04675. https://arxiv.org/pdf/2003.04675.pdf

[42] Chauhan N. (2020). "Decision Tree Algorithm Explained". *KDnuggets*, **20**(**2**). Online tutorial: https://www.kdnuggets.com/2020/01/decision-tree-algorithm-explained.html

[43] Elia M., Gajek C., Schiendorfer A., & Reif W. (2021). "An interactive web application for decision tree learning". *Proceedings of Machine Learning Research*, **141**, 11-16. http://proceedings.mlr.press/v141/elia21a.html

[44] Manapragada C., Webb G. I., & Salehi M. (2018). "Extremely Fast Decision Tree". In: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (pp. 1953-1962). doi:10.1145/3219819.3220005

[45] Musen M. A. (2015). "The Protégé project: a look back and a look forward". *AI Matters*, **1**(**4**), 4-12. doi:10.1145/2757001.2757003

[46] Terziyan V., & Nikulin A. (2021). "Semantics of Voids within Data: Ignorance-Aware Machine Learning". *ISPRS International Journal of Geo-Information,* **10**(**4**), 246. doi:10.3390/ijgi10040246

[47] Katasonov A., Kaykova O., Khriyenko O., Nikitin S., & Terziyan V. (2008). "Smart Semantic Middleware for the Internet of Things". In: *Proceedings of the Fifth International Conference on Informatics in Control, Automation and Robotics* (vol. 1, pp. 169-178). doi:10.5220/0001489001690178

[48] Puuronen, S., Terziyan, V., & Tsymbal, A. (1999). "A Dynamic Integration Algorithm for an Ensemble of Classifiers". *Lecture Notes in Computer Science*, **1609**, 592-600. Springer, Berlin, Heidelberg. doi:10.1007/BFb0095148

[49] Terziyan, V. (2008). "SmartResource–Proactive Self-Maintained Resources in Semantic Web: Lessons learned". *International Journal of Smart Home*, **2**(**2**), 33-57. https://gvpress.com/journals/IJSH/vol2_no2/3.pdf