Assessing model requirements for explainable AI: a template and exemplary case study

Michael Heider, Helena Stegherr, Richard Nordsieck, Jörg Hähner

Angaben zur Veröffentlichung / Publication details:

Heider, Michael, Helena Stegherr, Richard Nordsieck, and Jörg Hähner. 2023. "Assessing model requirements for explainable AI: a template and exemplary case study." *Artificial Life* 29 (4): 468–86. https://doi.org/10.1162/artl_a_00414.

Assessing Model Requirements for Explainable AI: A Template and Exemplary Case Study

Michael Heider*
Universität Augsburg
Organic Computing Group
michael.heider@uni-a.de

Helena Stegherr
Universität Augsburg
Organic Computing Group

Richard Nordsieck
XITASO GmbH
IT & Software Solutions

Jörg Hähner
Universität Augsburg
Organic Computing Group

Abstract In sociotechnical settings, human operators are increasingly assisted by decision support systems. By employing such systems, important properties of sociotechnical systems, such as self-adaptation and self-optimization, are expected to improve further. To be accepted by and engage efficiently with operators, decision support systems need to be able to provide explanations regarding the reasoning behind specific decisions. In this article, we propose the use of learning classifier systems (LCSs), a family of rule-based machine learning methods, to facilitate and highlight techniques to improve transparent decision-making. Furthermore, we present a novel approach to assessing application-specific explainability needs for the design of LCS models. For this, we propose an application-independent template of seven questions. We demonstrate the approach's use in an interview-based case study for a manufacturing scenario. We find that the answers received do yield useful insights for a well-designed LCS model and requirements for stakeholders to engage actively with an intelligent agent.

Keywords
Rule-based learning, self-explaining, decision support, sociotechnical system, learning classifier system, explainable AI

1 Introduction

Increasing automation of manufacturing creates a continuous interest in properties commonly associated with lifelike or organic computing systems, such as self-adaptation or self-optimization, within the production industry (Permin et al., 2016). These properties are often achieved using data-driven and learning methods (Lughofer et al., 2019; Schoettler et al., 2020; Zhang et al., 2017), as with increasing digitalization and Internet of Things (IoT) efforts, where more and more devices are interconnected and partake in complex problem solutions, data can be collected in large amounts. In modern factories, products are usually inspected by the machines' operators or specialized quality assurance personnel to assess their quality (see Figure 1). For the sake of simplicity, we subsume both roles under the term *operator*. Recent advances in automated inspection often integrate computer vision-based approaches (Margraf et al., 2017). However, these can be of limited use when quality is not assessable from the surface, for example, structural or chemical properties that involve laboratory testing. Thus these systems currently can only partially automate inspection, while the conclusions with regard to machine reconfiguration are still reached manually in many cases. This requires a large amount of operator knowledge and experience to achieve optimal or even satisfactory results. In settings with heterogeneous machines and few operators, the strain on operator

* Corresponding author.

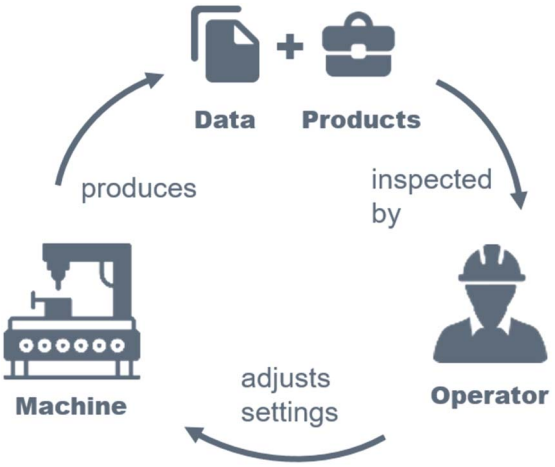

Figure 1. Operator-in-the-loop in modern manufacturing.

experience is further increased, and production can be seriously threatened by a loss of qualified personnel, for example, through retirement.

To reduce reliance on specific knowledge of operators and improve the self-adapting and self-optimizing systems, the operator can be assisted by decision support systems. These can easily incorporate large amounts of information simultaneously and are less biased to well-known settings, especially compared with operators that have only limited understanding of or experience with the machines. Such decision support systems utilize learning from past experience and ongoing human expert feedback. Combining human operators and supervised learning (SL) agents that collaboratively adjust machines (or lines thereof) that manufacture products expands the sociotechnical system with a collaborative decision-making dimension (see Figure 2).

Typical shop floor environments will feature many workers operating many machines, but not necessarily in a one-to-one array, for example, multiple workers might be needed to operate a single machine, while multiple other machines can be operated by a single worker due to automation. Additionally, to utilize the available data most efficiently, not every machine should need its own model, but models should generalize over multiple machines of the same or similar type. For production lines where multiple models would participate, the parameterization choices of preceding machines would need to be accounted for by subsequent models, for example, through the help of models of higher abstraction. In this environment, each individual model takes input from and advises multiple operators, while each individual operator might interact with different models throughout a shift.

An integral element for implementing these systems is that operators are able to trust decisions made by their recommendation agents. This requires the system to be self-explaining in both adequate form and abstraction level. However, when form and abstraction level can be considered adequate is highly use case specific and may also be user specific (see question 2 of section 4) (Belle & Papantonis, 2021; Herm et al., 2022). It involves an explanation regarding the basis of the recommendation, for example, what input parameters led to the output, as well as an assessment of the quality of the decision, for example, the expected error in quality when executing the recommended parameterization. In this article, we posit that learning classifier systems (LCSs) are well suited to be used within the proposed SL agent by reviewing different explainability techniques in light of this setting (see sections 2 and 3). We then introduce a template of research questions that need to be addressed to successfully apply LCSs (or other rule-based systems) in this context. We demonstrate the successful use of those questions in a case study in which we utilize them in a sequence of interviews with stakeholders from a producing company, the REHAU SE.

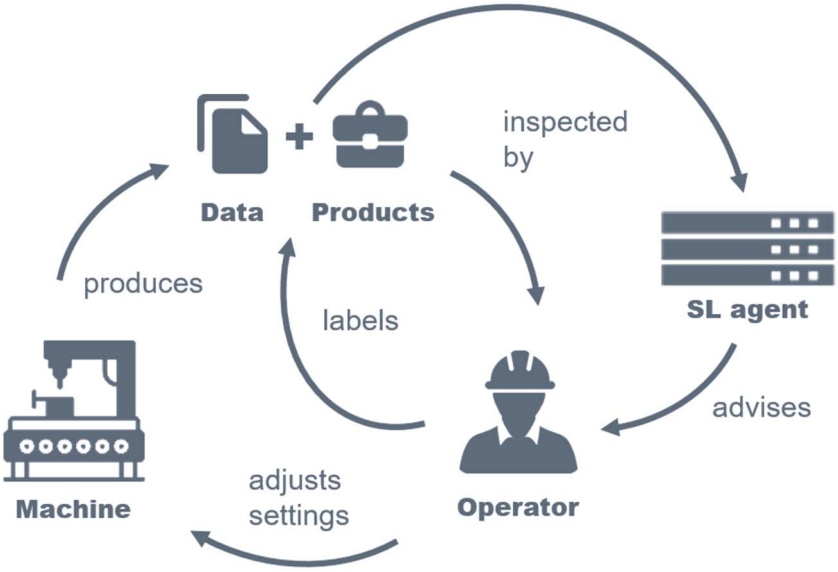

Figure 2. Assisted production using an agent trained with supervised learning (SL) during operation.

2 Learning Classifier Systems

LCSs are a family of rule-based learning systems (Urbanowicz & Moore, 2009). Though LCSs compose a diverse field, they share some common properties. In general, LCSs produce models consisting of a finite number of if-then rules (traditionally often referred to as *classifiers*), where individual premises (*conditions*) and, by extension, the global model structure are optimized using a—typically evolutionary—metaheuristic and the *conclusions* of the rules use a problem-dependent model. These classifiers or local models can then individually be ascribed a quality of their prediction within their respective subspace of the global model's input space. In our view, this structure is sufficient to motivate their application within a decision support system. However, we acknowledge that, as there is a plethora of possibilities to train such a model, choosing the "right" LCS for an actual implementation needs to be done case specifically, as some LCSs will yield better-performing models than others, and their transparency varies (for more details on transparency of models and related concepts, see Bacardit et al., 2022; Barredo Arrieta et al., 2020).

Explainability of machine learning models is usually differentiated into (a) *transparent methods*, allowing interpretation of decisions and comprehension of the model based on the model structure itself, and (b) *post hoc methods*, utilizing visualization, transformation of models into intrinsically transparent models, and similar techniques on models that are not by themselves transparent (Barredo Arrieta et al., 2020).

As rule-based learning systems, LCSs generally fall into the domain of transparent models and are regarded as excellent for interpretability because of their relation to human behavior (Barredo Arrieta et al., 2020). However, several factors can limit the degree to which humans can easily comprehend the model and follow its decision-making process. Most notable are the number of classifiers and their formulation. Conditions of rules in complex feature spaces are harder to understand than those that operate directly on the data, for example, higher-level features aggregating multiple sensor readings versus the readings themselves. Additionally, conditions can be formulated using nonlinear functions rather than readable decision boundaries (Bull & O'Hara, 2002). Conclusions of rules that utilize complex black box models, such as neural networks (Lanzi & Loiacono, 2006), are also harder to understand than linear or constant models, even if these local black box models

are usually much smaller than a model of the same class that encompasses the complete problem space would need to be.

These issues can warrant design adjustments within the LCS or the application of post hoc methods. The number of rules can be combated depending on the type of system considered: For Pittsburgh-style systems, this is usually achieved by promoting small individuals through adjustments of the fitness function (Bacardit & Garrell, 2007), whereas in Michigan-style systems, rule subsumption and compaction methods are applied (Liu et al., 2019, 2021a; Tan et al., 2013). An improved understanding of singular classifiers can be pursued by promoting simplicity during training through a suitable fitness function; by applying analyses typical for the respective models, for example, feature importance estimations in neural networks; and with a variety of visualization methods (Liu et al., 2019, 2021b; Urbanowicz et al., 2012).

3 LCSs in Industrial Decision Support Systems

Many different LCSs have been proposed over the years, and although originally envisioned as a powerful reinforcement learner, they have been extended for all learning paradigms (Urbanowicz & Moore, 2009). We consider the application as a decision support system that proposes settings to an operator and informs them of the reasoning behind this choice to be an SL task. This can be solved with either online or offline learning, as long as the model used to make recommendations provides a compacted version of itself for inference and subsequently serving explanations. The LCS learns from experiences, including sensor readings, product information, used machine settings, and resulting quality measures, all of which will be a mixture of real and categorical values. When tasked with assisting an operator, the SL agent uses sensor readings and product information to propose machine settings and predict the expected quality.

Besides the previously introduced explainability techniques, LCSs also easily allow us to provide operators with all examples from our training data that formed the local model (as we know which examples were matched by the rule's condition). This can help further the trust that the model's predictions are actually based on existing expertise. Going beyond traditional explaining by example (Barredo Arrieta et al., 2020), each example that influenced an individual rule's weights could hypothetically be listed, whereas in black box models, usually the entire sample influences every weight.

In Michigan-style LCSs, each individual rule gets ascribed a quality measure (or multiple thereof in XCS(F)). This (or in case of multiple measures, at least one of them) represents the rule's fitness and is used to guide an evolutionary process. Moreover, we can utilize these measures to provide our operator with additional information on how exact and therefore useful a recommendation is. Rules with a low prediction quality and thus a high expected error might provide poor machine settings, while other rules in the model might actually provide very useful settings. This disparity in different parts of the feature space can also allow insights into where new sampling should take place (Stein et al., 2017) and allows further differentiation of the model. Even if—viewed globally—the model is less than optimal, it can still be used within the SL agent and aid operators on problem instances where it is well fitted.

4 A Template to Assess Explainability Requirements and LCS Model Design

Following this examination of the applicability of LCSs as decision support systems for the parameterization of industrial machinery in a complex sociotechnical environment, we want to raise several questions that—in our view—need to be answered on a case-by-case basis. We assume that some parallels will exist between applications; however, it seems unlikely that general answers will hold for all or even the majority of cases. Note that we broaden the scope from our operators that interact directly or indirectly with the machine to all stakeholders that have a vested interest in the operation

of the shop floor, both digital and analog. Thus this could also include regulatory bodies, safety officers, engineering, management, customers, data scientists, and others.

We hypothesize that the seven following questions allow those responsible for model development and deployment to gain valuable insights into what is actually requested by those affected by such a model. Ideally, the answers are so detailed that exact requirements could be made on the design of a LCS model, which in turn would allow the design of suitable training algorithms. But even if the answers are not detailed or specific enough to make those decisions directly, they might serve to decide which algorithms or model designs might be suitable or whether LCSs are even the right choice. In cases in which explainability is not deemed important, practitioners can default back to deep learning. In other cases, decision trees or simple linear models might be more appropriate. The questions should also serve to determine whether these different models are more fitting for the use case and can also give insights into their design requirements. Furthermore, we assume that those questions allow stakeholders to engage what they have to expect from such a model and may later on be used to justify and explain certain limitations and trade-offs that have to be made without going into too much algorithmic detail. We intend for the questions to serve as a template for other practitioners in designing their own studies, if needed, by adding further questions or detailing and adjusting some of the existing ones. However, on the basis of our experience (among that of others detailed in section 5), we assume that they are fitting for many situations.

1. **Q1: To what extent does a stakeholder request explanations?** This can have numerous dimensions, such as depth, frequency, or diversity of explanations. Someone that operates the machine directly might prefer examples of past experiences, whereas quality assurance personnel might prefer visualizations, or vice versa. In this question, we assume that stakeholders may seek explanations that go beyond regulatory requirements, although a potential answer may be that they are not interested in further or deeper explanations. This raises another aspect: How important is explainability deemed if prediction quality potentially suffers?

2. **Q2: What are the differences within a type of stakeholder?** Tying directly into the previous question, we assume that the diverse stakeholders of a given type will answer questions regarding explainability differently. Individual stakeholders may also hold different understandings of the machine itself, so explanations would need to accommodate specific levels of prior knowledge. Furthermore, diversity between individual operators might be substantial and warrant personalization approaches.

3. **Q3: How many rules may the served model contain before being considered too large?** Smaller rule sets are easier to generate a general understanding of, while larger rule sets can provide a more diverse coverage of the input space and, therefore, potentially more accurate predictions. In some cases, such as explanations for specific decisions, the entirety of the rule set might not even be of interest, and stakeholders may prefer explanations to be limited to the rules whose conditions matched the situation.

4. **Q4: What form can conditions take before they are too complex to be understood?** Many rule representations have been proposed in the past, and although ellipsoids or neural networks can provide improved results, hyperrectangles (with a simple interval for each input dimension) might be easier to comprehend. Typical decision trees and random forests use hyperrectangular conditions with nonoverlapping feature space partitions. Beyond those options is the LCS-specific concept of code fragments, a program tree–inspired way of capturing nonlinear decision boundaries (Iqbal et al., 2014). They are fairly human readable, albeit less than hyperrectangles, in comparison to neural networks. This should also probe whether the exact condition is even considered relevant or if operators are content with knowing that it applies to a certain instance. However, counterfactual-based

explanations might be a worthwhile effort enabled by clear decision boundaries that can be understood by humans. That is, if the situation were slightly different, another rule might apply, changing the model's prediction.

5. **Q5: How important are explanations of why the decision boundary of a rule is placed a certain way?** In LCSs, the model structure (and decision boundary of each rule) is optimized using a metaheuristic to localize the rules in a way that they fit the data well. Within this question, we want to ascertain how important insights into this process are to operators.

6. **Q6: What form can conclusions take before they are too complex to be understood?** While linear models are widely regarded as easily comprehensible, more complex models might yield better results, and typical explanations, such as feature importance analysis, may satisfy the stakeholders' want for understanding the decision-making process. This also translates to the use of mixing models (where multiple rules are used to construct a prediction) and the comprehension thereof.

7. **Q7: What information do stakeholders request about the training process?** This question aims toward training in general and the steps that are performed in the process of deriving a model rather than toward analyzing the utilized model. An important aspect of this can be gathering, cleaning, and selecting data and responsibilities therein.

Regarding the specific model (and algorithmic) design decisions practitioners can base on the outcome of those questions, we want to highlight section 5.6. In general, questions Q1 and Q2 serve mostly to determine the trade-offs of explainability and performance and the differences between stakeholders and individuals. They might tell us that we should train and deploy individual models for optimal acceptance and stakeholder satisfaction. They might also highlight the need for different visualization and analysis tools for the models or the form in which explanations should be given, regardless of model design. Q3–Q6 are more specific for rule-based systems like LCS and decision trees. These also give the most insights into what the deployed model should look like. Q3 gives the relevance of rule set size in the optimization process (for Pittsburgh-style systems, directly during training, and for Michigan-style systems, in posttraining compaction). Q4 answers directly what condition scheme rules should use, whereas Q6 answers the same for the rules' predictive model. Both decisions are usually made before training is started. Q5 determines whether the training algorithm itself should be explainable (or to what degree this is needed). There are some approaches into making the stochastic optimization of evolutionary algorithms explainable (Bacardit et al., 2022), albeit not specifically focused on LCS training. Q7 is again more general and focused, not on LCS model design, but rather on the conditions surrounding training, for example, who (individuals, departments) was involved or what data were used (and, importantly, what data were not).

5 Case Study: Assisting Operators in a Chemical Industry Plant

To demonstrate a potential use of our proposed template of questions to determine the requirements for a self-explaining sociotechnical system that supports operators in their day-to-day tasks, while also satisfying other stakeholders' needs, we performed an interview-based case study. In this case study, we interviewed a variety of different stakeholders about their individual needs, as well as their colleagues' and subordinates' needs, in such a system before its final design and implementation. Note that this study serves as an example of how to apply our proposed questions, and their answers will likely be very use case specific and might not be transferable to other use cases. This issue of nongeneralizing answers is very typical for similar studies regarding explainability needs (Belle & Papantonis, 2021; Herm et al., 2022). Therefore we have to work with small sample sizes

(as few individuals in a specific stakeholder role exist) and cannot apply many of the quantitative analysis tools that might be available for large-scale studies. The envisioned operator assistance system (OAS) is to be employed in an international chemical industry company, the REHAU SE.[1] REHAU plans on piloting it in a German plant of its interior solutions branch, which is the main focus of our case study, where so-called *edge bands* are produced. However, we also interviewed a stakeholder from a plant of their window solutions branch to broaden the scope, potentially find differences even between branches of a single company, and, we hope, find some answers that can be applied to other branches in the future.

5.1 Operator Assistance System

The primary motivation behind the OAS is to disencumber operators and reduce their overall workload, which currently is substantial. This is to be done through increased automation of, currently manual, routine adjustments and by providing operators with more insights into disturbances and with potential solutions. Overall, this increases the robustness of the production and reduces material and energy waste.

In the line control, OAS-like systems assist operators with manual configuration of individual machines in the line or overarching parameters and partially automates it. Its components are largely well understood from a chemical engineering point of view. Although, arguably, some level of explanation to operators could always be beneficial, these algorithms do not employ any form of machine learning component and therefore fall out of the scope of our study, where the focus is an SL agent operating as one of the, potentially many, systems forming the overall OAS. Another component currently in production is a tool that aggregates existing knowledge in an easy-to-navigate, tree-like structure. When encountering some issue, for example, a quality defect, the operator navigates via web interface from broad areas to specific defects or disturbances, where individual stages are described both textually and visually. Once the issue is narrowed down, the operator is presented with common solutions to the problem and an estimate of how successful these have been in the past. After the issue is resolved, the operator is asked to give feedback on whether the provided suggestion was helpful and correct, promoting this suggestion for the next operator to encounter this issue. These paths through the tree-like structure can be reformulated as rules that can potentially in turn be used inside a LCS, either as an initial population before training or by manual insertion into the trained model, where they serve to cover areas of the problem space where training examples were too scarce to create sufficiently accurate rules. Additionally, these rules can be used for potential explanations of evolved rules, as they should—due to their crowd-sourced nature—be deemed more reliable by operators than some rather high-level and maybe opaque machine learning process. For simplicity's sake, we refer to the envisioned SL-based agent as part of the OAS as the *agent* in the remainder of this text and primarily consider its specific requirements without limiting other components.

Depending on its maturity, predictive power, and stakeholder trust, the agent can be employed at different levels:

1. Predict the quality of a machine parameterization selected by the operator.
2. Actively make suggestions for possible parameterizations and their predicted product quality to the operator.
3. Set a single parameterization and prompt the operator to confirm.
4. Regulate the process parameters fully automatically, for example, when product quality or process stability indicators drop, with the operator acting only as a supervisor.

1 https://www.rehau.com/.

These levels also change the operator's role in our sociotechnical system of machine, agent, and operator in that the higher levels lessen the mental load of trying to come up with possible solutions and transform the operator to an executor of physical adjustments and tasks while keeping them in a position of supervisory responsibility. Likely, different settings in which the agent is to be used will allow higher levels of operation earlier. In less crucial parts of a production line (i.e., those not sensitive or prone to significant damage), the agent will be able to choose from a wider range of still sufficient parameterizations while facing less scrutiny by different stakeholders. The same holds for areas with different data availability and quality. Ultimately, any SL prediction is dependent on diverse and correct data for training. Machines of a line that have long been digitized and fitted with well-calibrated sensors will more likely offer such data than machines that have until recently been controlled by analog means. For these newly digitized machines, it might even be unknown what sensors are missing to make meaningful predictions, and they might not yet have been online long enough to gather sufficient data or even to allow the determination of what noise is to be expected during operation, for example, the impact of seasonal changes.

Regardless of the specific scenarios, it is clear that to get such an agent into production, relevant stakeholders have to be on board from the early stages of its design process. Stakeholders also reflected this in early talks about potential use cases. In these talks, they first raised the, albeit expected, issue of transparency of such an agent and its decisions as central to generating enough trust to employ it.

5.2 Extrusion: An Example Application of the OAS and Its Agent

In the production of plastics, a typical first part of a production line is the melting of synthetic granulates (or powders) and subsequent form-giving extrusion of the heated semifluid mass. The correct pressure—and, for many products, also the temperature—is crucial to ensure sufficient dimensional accuracy and therefore product quality. The exact values are dependent primarily on size, shape, and material type, but from a process engineering point of view, it is very much possible to find a range of values that can be considered sufficiently optimal to guarantee the desired product quality. Operators will control for this measurable parameter rather than shape and size, as process engineering guarantees desired dimensionality whenever the correct pressure is applied. This also has one key advantage for prediction: The resulting learning task is a regression for which sensor readings are comparatively easy to obtain, whereas controlling a multidimensional shape and size vector, for which complicated and highly accurate laser scans would be needed, is much less straightforward.

In REHAU's interior solutions branch, specifically edge band production, extrusion pressure is regulated by eight adjustable parameters. Additionally, a multitude of additional sensor readings, primarily temperatures in different sections of the extruder, are available. The adjustable parameters show highly nonlinear relationships with the target, warranting sophisticated self-learning and—because of the requirements on transparency—self-explaining systems.

5.3 Study Design

One critical issue to be solved to actually get the agent into use in a scenario similar to the one presented in section 5.2 is stakeholder acceptance. This acceptance needs to be nurtured from the early design stages by making choices according to the wishes (and worries) of the various stakeholders. From early preliminary talks with R&D and different management levels, we already knew that whatever the exact embedding system design would be, the self-explainability of the employed agent would likely be central. This already hinted toward a LCS being a very plausible choice for the learning algorithm. Thus we use the template raised in section 4 with relevant stakeholders to determine if the assumptions that explainability is very important are even correct and, if so, how the resulting LCS model should be designed. This serves a second purpose: Discussing these issues with stakeholders in the form of questions allows them early participation in the design process, allowing the OAS to be developed according to their requirements. This reinforces the

perception of holding a stake rather than the feeling that some ill-suited system was forced upon them. In another direction, but complementary to the described goals, this also facilitates testing the validity and applicability of the questions raised and whether they even allow meaningful insights. This is an important consideration for potential future applications of the template (or if they turn out to be suboptimal for a reformulated version).

To validate the applicability of the questions and gain some perspective on what answers we can expect and where additional clarifications or input might be warranted, we conducted a pilot interview with the director of Smart Factory and selected members of his department, which is responsible for machine automation, data science, IoT, assistance systems, and sensors. We found that the questions can be used as proposed in section 4, but more explanation, especially of the specific nature of LCS models, is beneficial to get more useful answers for the LCS-specific questions. Importantly, we found that explanations are definitely desired on many levels. More results are discussed in section 5.4.1.

Our main study was conducted in individual interviews with stakeholders of about 45 min. As all participants were German and few work with English on a daily basis, these interviews were conducted in German. Interviews began with the interviewee prompted to give some information about themselves and their current job as well as their job history at REHAU. After a short introduction to the general topic, possible levels on which the agent can operate, and an example use case based on the extruder (see section 5.2), the stakeholders were presented with the seven questions and some additional explanations, examples, and follow-ups. The questions were also reformulated into German, and technical (machine learning) jargon was—where possible—kept to a minimum. As LCS (and other machine learning model types) were unknown to most participants, examples of a 1-D task solution and an 8-D example rule were also presented before Q3, where the number of rules is discussed. Interviewees were strongly encouraged to ask for clarifications if some point of a question was unclear and received additional context or details if they expressed trouble answering. The interview was aided by a set of slides, so interviewees could follow along and reread a question, if needed.[2]

The relevant archetypical stakeholder roles can be summarized as follows:

- *Operators* operate the machine to manufacture a product. Typically, operation takes place in a one-to-one ratio in the interior solutions branch and sometimes in a one-to-many ratio in the window solutions branch. They interact with the agent throughout their shift and, as they are responsible for smooth production, rely heavily on its capabilities. Especially (comparatively) inexperienced operators often need assistance, whereas seasoned (10+ years of experience) operators will rarely be in situations where they consult others.
- *Team leaders* supervise a group of operators on the shop floor on a given shift. For troubleshooting, team leaders are the subsequent responders when colleagues on the next line are unable to assist an operator. Therefore they interact with the line control (and thus the OAS and the agent) on a frequent basis. If even some of the operators' questions and issues get resolved by an agent, the team leader's job becomes considerably less stressful, whereas if the agent gives poor advice or confuses the operator, their job might become more difficult.
- *Production managers* are ultimately responsible for the entire plant's production and are thus very interested in past and projected manufacturing capabilities.
- *Process engineers* have the deepest knowledge of the underlying process. They have deep foundational understanding of maintaining process stability, which they are also constantly trying to improve. They operate either closely to or within the plant, where they are second in line for troubleshooting when operators and their supervisors cannot fix an issue at hand

2 These slides can be found at https://doi.org/10.5281/zenodo.6505010.

by themselves or in more centralized process engineering departments, where they determine set parameters and machine configurations for new material compositions and product types and perform other developmental steps toward machine improvement and innovation.

- *Data scientists* are expected to maintain, improve, and expand the capabilities and possible applications of the agent (and other machine learning methods in use). They select which data are to be used, determine what new sensors are needed, and validate the correctness of readings. From a model perspective, it is likely that the agent encapsulates multiple models that are directly trained to predict on a particular machine (model) or even for a specific product, rather than a singular generalized model that solves all tasks, although generalization is, overall, desirable, as fewer models are easier to maintain. Data scientists would thus need to determine which machines and products can share a model and for which combinations other models are needed. Ultimately, a badly performing agent is the responsibility of the data science team.

We want to add an important disclaimer for this specific study that is, however, very likely applicable to most similar studies: These stakeholder archetypes are often not clearly distinguished in a single person, and a person's view of certain aspects might be heavily influenced by their (job) history so that, despite their current position, they still express views we can clearly attribute to another archetype. This constitutes a form of bias that needs to be accounted for when drawing qualitative conclusions based on such studies. With large enough sample sizes, this might average out. However, in many typical industrial settings, the number of individuals in a given position may often be too small (Belle & Papantonis, 2021). Team leaders are often trained process engineers who have been operators at REHAU before undergoing additional education. In-plant process engineers often have a management role as well, with responsibilities for sections of the plant. Although, in this archetype specifically, the exact position of a person between R&D responsibilities, where university graduates are more common, and day-to-day operations widely varies. We still chose to present these as one archetype, as the general questions they ask of the agent are similar. The interview partners available for this study were selected to allow an overview of all roles, and interviewees were asked to distinguish between the different roles they might find themselves in or have held in the past for their answers. They were also requested to separately answer as operators and based on their perceptions of an operator's requirements.

5.4 Interview Findings

From the conducted interviews, we find a need for self-explainability of the envisioned agent and that simpler models are generally preferable. More detailed descriptions of the answers to the seven questions are given in section 5.4.1 for our pilot interview and section 5.4.2 for the main interviews that were conducted afterward. Reassuringly, we also found that the agent is indeed wanted.

5.4.1 Pilot Interview

In this first interview, we aimed primarily at validating the applicability of the proposed questions to gain insights regarding the envisioned scenario. It was conducted with the director of Smart Factory at REHAU in the presence of some members of his department who were involved in the already existing parts of the OAS and a variety of data science applications. The concept of a SL-based agent to assist operators and its various levels of application were well established previously. Answers, as given by the director regarding his perception of various stakeholders' requirements, were recorded and are stated in the following paragraphs.

1. **Q1: To what extent does a stakeholder request explanations?** For operators, this depends primarily on the agent's autonomy. The more autonomously the agent acts, the less the individual operator will request explanation. In contrast, the process engineer will

always want in-depth explanations. This requirement will likely increase with agent autonomy, for example, when debugging potential issues, as the operator will have less insight into what was configured and why. Team leaders will require more explanation and more depth than operators. Data scientists will want maximum transparency and self-explainability.

2. **Q2: What are the differences within a type of stakeholder?** For operators, the frequency and depth of explanation will depend highly on their experience. Experienced operators will probably disregard the agent completely and use their own knowledge to solve upcoming issues. Thus they will also not request any explanations. For other stakeholders, experience might matter for simple tasks—for example, if the prediction aligns with their mental model, they will not request an explanation—but overall explanations will be requested by all personnel in these roles.

3. **Q3: How many rules may the served model contain before being too large?** As LCS models and their structures' implications were not completely clear, we presented a small ad hoc visual aid of what a LCS model might look like, which we then also kept for the main interviews. Data scientists may be the only stakeholders who might want to analyze the model in its entirety. Other stakeholders, specifically operators and team leaders, will be interested primarily in the model's situation-specific predictions. Therefore explanations of the given mixing model will be more relevant, and the global model can contain a large quantity of rules, as long as it can still be experimentally or statistically verified, that is, through well-chosen test data. The mixing model should also contain few rules. This question also brought up a point about submodels: They should be trained in a way so as to directly determine the most important features or parameters for a given prediction, for example, by forcing two to three coefficients to be considerably larger than others in a linear model.

4. **Q4: What form can conditions take before they are too complex to be understood?** Interval-based rule conditions are strongly preferred over other options for both regression and classification tasks.

5. **Q5: How important are explanations of why the decision boundary of a rule is placed a certain way?** Operators and team leaders will probably not have this question and take the conditions as is. Some trace-back to the training sample might be interesting for data scientists but is not needed.

6. **Q6: What form can conclusions take before they are too complex to be understood?** This question was deemed impossible to answer without taking the model's task and performance into account. In general, simpler submodels are preferred.

7. **Q7: What information do stakeholders request about the training process?** The director was unable to confidently provide deeper insights into this question. Likely, information is of interest, but the respective stakeholders would need to clarify the exact levels.

5.4.2 Main Interviews

Following the pilot interview, an expanded introduction to both the possible application of the agent and a LCS was prepared. Additionally, the seven questions were translated into German, and where applicable, follow-up questions based on answers given in the pilot interview were formulated. After that, four interviews were conducted. As this group of stakeholders was quite heterogeneous with regard to different perspectives on the questions and operators' views, we attribute the (paraphrased) statements to the respective interviewees (A–D).

- A is currently a process engineer and supervisor with administrative responsibilities in edge band production. They started in the company as an operator and then became team leader before a promotion into their current position. They supervise and interact with operators and machines throughout a normal workday.
- B is from the window solutions branch and head of recycling and plant optimization. They started as an operator before training as a process engineer and receiving various promotions up to plant management. Therefore they have a good perception of all relevant in-plant roles and might already give some perspective if the answers can be reused for a similar manufacturing process for a different product at another plant.
- C is head of the data lab—a department responsible for all data management, analysis, and science. They have a strong statistics background and have been working with various stakeholders from multiple plants for years, including directly interacting with operators at the machines over long periods of time.
- D is a member of the data lab. Originally part of R&D, they subsequently joined the IT department and then—with its foundation—the data lab. They are responsible primarily for keeping data-related systems, such as the envisioned agent, running and up to date.

In this section, the answers of the interviewees regarding the various stakeholder archetypes are presented in a question-wise manner. Where conflicting answers were given, we present both.

1. **Q1: To what extent does a stakeholder request explanations?**
   - **Operator**. New operators are thankful for all assistance, including explanations (A). Explanations also enable them to fix issues on their own (A, B). In general, short textual explanations of two to three sentences are preferred (A, B). Probabilities of success of a proposed parameterization and rule quality could be useful but are not mandatory, as long as the model itself is not guessing (B). Explanations should be offered on request rather than by default on every prediction or reparameterization (B). They could be enriched with images of issues that may arise from the suboptimal parameterization or other information about past production (A, D). Graphs and dashboards are not useful for operators (A, B). Neither are mathematical formulas (A, B, C, D). As long as the performance is on some generous level of practical equivalence, transparent models are preferred over better-performing ones (A).
   - **Team leader**. In addition to textual explanations, graphs can be useful in understanding and improving the manufacturing process (A). However, as long as production proceeds as scheduled, team leaders might not care for explanations (B). Not-as-explainable models with better performance can still be useful (A).
   - **Production manager**. The main interest is with keeping production up and efficient (B). Understanding why errors are occurring is of deep interest so as to prevent them in future (B). In addition to textual explanations, which are likely too low level for most situations in which management is involved, high-level dashboards and graphs allow them to understand their production (B). Model transparency is more important to them, but in the end, pragmatism reigns (B).
   - **Process engineer**. Being tasked with both ad hoc debugging and long-term improvements, process engineers have a deep interest in understanding the manufacturing process (A). Machine learning models that may infer connections from data that are unknown or at least unquantified by humans are of great relevance to achieving their goals (A). However, to be analyzed, these models need to be as transparent as possible (A). Diverse tools for in-depth explanations are

very important (A). Process engineers might not analyze every decision but rather all that went wrong, as well as the general model (A).

- **Data scientist**. Ideally, the model would be a complete white box, as transparency and explanations are preferable (D). However, a substantially worse white box model should be replaced with a gray or black box system that undergoes a rigorous statistical analysis (C, D). A well-validated model that can be inspected via graphs and dashboards could be deployed even without inherent transparency (C). Depending on the task, transparency could also be approximated via post hoc analyses, although this would make the usability for other stakeholders questionable, depending on the correlation between the original black box and its transfer-learned pendant created through post hoc analysis, such as LIME (Ribeiro et al., 2016) (C). Explanations should be in depth and may include mathematical formulas (C, D).

2. **Q2: What are the differences within a type of stakeholder?** For all stakeholder archetypes, substantial experience will result in some predictions and decisions being obvious, thus not requiring explanation (A, B, C, D). Data scientists might still want to understand how the model inferred this from data, but this would not warrant a self-explaining model (C). Less experienced stakeholders will often require more or more in-depth explanations than those with average experience, although on the other hand, very experienced stakeholders might in turn require more depth to be convinced or to understand how the model found something they did not (A, B, C). Whether explanations are requested is mostly dependent on attitude and motivation rather than experience (A, B). The broadest spectrum is shown within the operator role (A, B). For inexperienced operators, consulting the system replaces disturbing their colleagues and/or supervisors to ask for their help, which will increase the agent's acceptance (B). Personalization of explanations might be good for individual operators but would greatly complicate the team leaders' and process engineers' user experience whenever they are called for assistance (A).

3. **Q3: How many rules may the served model contain before being too large?**

   - **Operator and team leader**. The overall model's number of rules can be as complex as needed; however, in a given situation, only a few (up to 4 [A]) may match and be included in the mixing model (A, B, C, D). Additionally, rules should ideally be limited in a way so as to promote high weights for only three to four features at most, with other features having considerably smaller weights (A, D).
   - **Process engineer**. A process engineer will often analyze the full model and therefore require it to be small (A). However, the exact size is problem-dependent (A). For the extrusion problem, 15–20 rules should be an upper limit (A, D).
   - **Data scientist**. Matching rules are more important than the totality of rules (C, D). Intense validation of a subset of rules will likely allow data scientists to trust the other rules, as long as they share performance metrics (C). Overall, rule similarity is also important in that many dissimilar rules are more acceptable than high overlaps (C). However, upon further probing, sizes of 30–100 rules were deemed as indicating highly complex models for successful analysis (C, D).

4. **Q4: What form can conditions take before they are too complex to be understood?**

   - **Operator**. For operators, the specific condition does not need to be analyzed, as long as we can assure that this rule does apply (A, B). However, when interacting with operators to explain certain decision-making processes, complex models might make this more difficult (D).

- **Team leader and process engineer**. Easier to analyze is preferable (B). They should not be more complex than intervals (A).
- **Data scientist**. More complex conditions should be possible, as long as they undergo post hoc analysis, for example, LIME (C). If the LCS has proven to produce well-placed decision boundaries for similar problems, not all rules of every model would need to be analyzed (C). For practically equivalent performance, easier conditions are strongly preferred (D). With a higher degree of automation, analyzing the condition becomes more important (D).

5. **Q5: How important are explanations of why the decision boundary of a rule is placed a certain way?** The interviewees were in agreement that there is no need for explanation why the trained model exhibits certain decision boundaries and how the optimizer found these. The data scientist might have an interest in the process from a scientific point of view, but for machine operation and operator assistance through a trained model, this is not relevant (C, D).

6. **Q6: What form can conclusions take before they are too complex to be understood?**

- **Operator**. Operators will likely not care about specifics, as long as a textual explanation for the central aspects (e.g., feature importance/influence on prediction) is given (B). Models should be linear (A). An analysis of the mixing of the currently matching rules is sufficient (A).
- **Team leader and process engineer**. Submodels should be kept as simple as feasible (B). Ideally, submodels are linear (A). In addition to an in-production use, process engineers will want to analyze the model(s) to improve the process itself, for example, through changes in hardware, and for this, the models need to be understandable to them (D).
- **Production manager**. Individual predictions are less important than overall system performance (D).
- **Data scientist**. The use of more complex submodels should be possible (C, D), although simpler models are always preferable (C). If complex models are used, they would need to undergo rigorous individual testing and analysis (C). However, for better-performing submodels, this would be worth it (C, D).

7. **Q7: What information do stakeholders request about the training process?**

- **Operator**. With more available information, the operator's trust in the predictions will be higher (A). They care about which lines and which products were used for information gathering and by whom features were selected and models were built (A). As long as predictions are correct, operators will take the suggestions as is and not further request such information (B).
- **Team leader**. More detailed information than for operators as well as some form of involvement in the design process are requested (A).
- **Production manager**. Some information on a high abstraction level is sufficient, for example, where the responsibilities lie (B). Deployment time, lifetime performance, and possible adaptations based on products and performance are relevant (D).
- **Process engineer**. To determine if model performance is in line with the current understanding of the process, and to subsequently improve process stability on the basis of the model's production, engineers require as much information as available

(A, B, C). They should also be involved early on to avoid model biases from possible correlations without causation within the data (C).

- **Data scientist**. Although multiple stakeholder archetypes will request all information available, more than anyone else, data scientists will want to do statistical testing and analysis of the models (C). They will analyze train–test splits in detail (C). With the model in production, they employ statistical measures to detect possible concept or sensor drifts (C).

5.5 Summary

We find that stakeholder archetypes have—at times substantially—diverging requirements toward the explainability of the model. All stakeholders would prefer transparent models, as long as performance is practically equivalent. However, should this not be the case, the desire for model transparency depends highly on both the archetype and the individual person.

Within the group of operators, some might not ever consult the agent, and many might not care for its explanations, as long as predictions—or derived parameterizations—are correct. Regardless, substantial numbers of operators will both follow the agent's suggestions and check its explanations. These explanations can serve two purposes. On one hand, they help operators check for plausibility of a decision based on their own mental model and therefore increase trust in the agent. On the other hand, they may update an operator's mental model, which is especially important for newer and inexperienced operators who would otherwise need to rely on a colleague's or supervisor's assistance. Regarding LCSs, operators tend only to want to analyze currently matching rules. These should be few in number and kept as simple as possible. Operators want explanations primarily in a short textual form, ideally directly generated from those rules. With this role especially, we found staunch differences between the two plants, where interior solutions' operators want explanations much more frequently than do their window solutions' operator counterparts, where explanations are likely only requested in cases of production issues and defects. Although these might be attributable to the interviewees, it is very plausible that differences in the manufacturing process and how machines are interfaced within the sociotechnical system are the root of diverging answers, for example, the fact that within window solutions, a single operator operates multiple lines.

Team leaders largely follow the trends set by their direct subordinates, the operators. However, because of their increased responsibilities, they require more, deeper, and more diversely represented explanations from easy-to-analyze models. Again, the two plants seem to differ with regard to frequency of explanations for this role, although the trend is less substantial.

Production managers will less frequently interact with the agent and primarily require information about its performance and, if that is poor, will request more information into likely reasons. The agent could, for example, explain its poor performance in certain areas of the feature space with poor sampling, high noise, or unexpectedly complex parameter–target relationships. Individual decisions are unlikely to be analyzed by production managers. However, depending on their background, from a personal standpoint, they might be quite interested in what is running in their plant and how it works.

The process engineer requires the most in-depth and diverse explanations and general model analysis capabilities. From our interviews, we found a second aspect of use for the models besides their application within our operator assistance setting, namely, to analyze the models (or the agent in its entirety) to deduce process improvements that go beyond a parameterization. This can range from the hardware setup itself to chemical mixtures of line inputs to hydraulic valve switching. The simplest models are strongly preferred for both aspects. Process engineers are less diverse in their requirements, both from an individual and from a plant-wise perspective. In contrast to the director of Smart Factory's perception, not only data scientists but also process engineers will want to analyze the full model.

Data scientists were overall relatively open to deploying gray box or even black box models, as long as they had undergone substantial statistical verification or been made explainable through

post hoc analyses. However, transparency is preferable, as statistical verification of a black box model can be sufficient if one deeply understands the statistical decision-making process and possible fallacies therein but is hard to convey to stakeholders who do not have such knowledge and training.

By gaining an understanding of the various stakeholders' requirements through this study, we also validated that the seven questions are useful to determine them. We found that differences between the two plants seemingly exist for some but not all of our identified stakeholder archetypes. Likely, different domains and companies will also yield slightly different answers. Thus these questions should serve as a template for how to design specific studies. Additionally, we found that potential users not only want to be included in the agent design process but also have important uses for the agent and its models that are not included in the originally envisioned case but that can also be solved without a differently or separately designed system and that did not come up in previous discussion.

5.6 Consequences for Model Design

Whereas the previous section highlighted the most important interview findings with regard to both likely interactions with agents and their respective models and the corresponding high-level explainability requirements, for example, how to present explanations to operators versus team leaders, we want to focus in this section on some more technical aspects regarding model (and algorithmic) design. We want to stress again that the template of questions focuses primarily on the returned model after training completes.

Although Q5 did ask about the training process itself, we found that—in the presented use case—it is largely irrelevant for the comprehensibility of an OAS agent. Therefore the LCS's algorithmic design can be freely chosen. As long as the resulting model fulfills certain criteria, it is irrelevant whether practitioners use a Michigan-style, Pittsburgh-style, or hybrid approach. The same goes for the deeper details, for example, how model updates or credit assignments are made, the fitness function, when to use subsumption, the specific compaction technique, how to cover, or what evolutionary algorithm to apply. None of the stakeholders expressed deeper interest in detailed knowledge about those mechanisms for the use case in the OAS. Data scientists are interested in this from a professional point of view (and are likely the ones ultimately making those decisions) but stressed the importance of the model rather than the training process. Not having to derive explanations of the behavior of evolutionary algorithms is also greatly advantageous for practitioners, as this is a challenging open issue (cf. Bacardit et al., 2022).

Though the algorithmic side of training was not confined by requirements, the expected structure of a trained model is. This does of course have indirect repercussions for the algorithmic properties of the LCS and how it should be configured to best arrive at such a model. The more straightforward aspects are the wishes for interval-based (hyperrectangular) conditions and linear models (for regression). Both aspects are typically defined before training. For the linear models needed for regression, we found that some confinements should be made regarding their coefficients, as stakeholders would prefer it if few (i.e., approximately three) of those were large at a time to more easily pinpoint influences. Coefficient control is a very common requirement for machine learning models and is typically solved by some form of regularization, for example, the well-known Tikhonov regularization, to keep coefficients small. How to design this component in a LCS depends on how the updates are made. In online-learning (e.g., Michigan-style) systems, this is less straightforward than for batch learning–based (e.g., Pittsburgh-style) systems, as in those systems, the traditional regularizations can be used directly. An easy-to-implement option for Michigan-style systems would be to use the returned rules' matching functions but retrain their linear model's coefficients in the same way. A very important aspect for some stakeholders is the quantity of rules—both the total number of rules and how many rules partake in a prediction. The total number of rules is controlled via subsumption and compaction in Michigan-style systems and directly in the fitness function and by pruning techniques in Pittsburgh-style systems. For fitness-based control, a practitioner could assign rule sets over a certain size an arbitrarily low fitness. For compaction-based

control (which is applied after training completes), an algorithm has to be chosen that offers an option to shift the performance–size trade-off in a certain direction. One straightforward option would be to use a binary genetic algorithm for that and again head toward fitness-based control. How many rules partake in a prediction is defined by the mixing model. A very common mixing-model approach is to use a fitness-based weighted average of the prediction. Here it would be quite simple to adapt the mixing model to take only the top n (e.g., three) rules according to the fitness into account. As this might change where rules should be placed to make smoother decision boundaries, we strongly recommend using such mixing restrictions in training rather than afterward.

Overall, although we find that practitioners still have large amounts of algorithmic freedom to train their models, we can make some clear restrictions to the expected final model structure.

6 Conclusion

In this article, we introduced a sociotechnical system within an industrial manufacturing setting where operators and SL agents could collaboratively adjust machine settings to optimize product quality. In these systems, operators can interact with a variety of heterogeneous machines and different agents throughout a single shift, while the agents also interact with different operators. Assisting the operators with recommendations from the agents decreases the necessity for experience and helps extract and conserve the experience of senior operators that might otherwise be lost over time. We reintroduced LCSs and reviewed why these rule-based systems are generally considered explainable. Building on that, we expanded on possible requirements for the design of a LCS within our agent and highlighted beneficial properties of LCSs for this application. This led to the formulation of seven guiding questions to assess the explainability requirements of individual use cases. Three of these questions are applicable to a variety of machine learning models—for example, *To what extent does a stakeholder request explanations?*—and aim at analyzing general wants and needs, whereas the other four questions are more specific to rule-based systems (LCSs, decision trees, etc.). We expect that answers to these questions are domain- and stakeholder-specific and would need to be answered independently for each setting or even each use case of similar agents. To demonstrate the validity of the questions for gaining relevant insights, we conducted a case study at the REHAU SE company. We interviewed stakeholders from two manufacturing plants and the centralized data science department, which offers solutions to all plants within the company. We anticipate that an operator assistance system utilizing machine learning would likely improve production. Predictive and decision-making systems should exhibit self-explainability, although the variety, frequency, and depth depend on stakeholder roles as well as individual investment. For some stakeholders at the second plant—even within the same company—differences were noticeable. Thus we got indications that a use case–specific reiteration of the questions will be required. Overall, the seven guiding questions do yield useful insights, and it is possible to design a system based on these answers. Whether such a system does actually completely fulfill all requirements—as they might not have been voiced—and if it will be well received will be answered in the future. Although we assume that general trends should be transferable, these questions and answers serve as a template whenever applying rule-based learning systems to a new scenario where comprehensibility is essential, and we invite other researchers to utilize this template. Consequently, we are confident that LCSs can introduce self-explainability into these sociotechnical-systems while advancing industrial manufacturing practices.

Acknowledgments

This article is an extension of previous work presented as part of LIFELIKE 2021, colocated with the 2021 Conference on Artificial Life (ALIFE 2021). We want to thank the REHAU SE for its elemental cooperation with this work by providing the interviewees for the case study. This work was partially funded by the Bavarian Ministry of Economic Affairs, Regional Development, and Energy and the Deutsche Forschungsgemeinschaft. The open access publication of this article was supported by the University of Augsburg.

References

Bacardit, J., Brownlee, A. E. I., Cagnoni, S., Iacca, G., McCall, J., & Walker, D. (2022). The intersection of evolutionary computation and explainable AI. In *Proceedings of the Genetic and Evolutionary Computation conference companion* (pp. 1757–1762). Association for Computing Machinery. https://doi.org/10.1145/3520304.3533974

Bacardit, J., & Garrell, J. M. (2007). Bloat control and generalization pressure using the minimum description length principle for a Pittsburgh approach learning classifier system. In T. Kovacs, X. Llorà, K. Takadama, P. L. Lanzi, W. Stolzmann, & S. W. Wilson (Eds.), *Learning classifier systems* (pp. 59–79). Springer. https://doi.org/10.1007/978-3-540-71231-2_5

Barredo Arrieta, A., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., Garcia, S., Gil-Lopez, S., Molina, D., Benjamins, R., Chatila, R., & Herrera, F. (2020). Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion*, *58*, 82–115. https://doi.org/10.1016/j.inffus.2019.12.012

Belle, V., & Papantonis, I. (2021). Principles and practice of explainable machine learning. *Frontiers in Big Data*, *4*. https://doi.org/10.3389/fdata.2021.688969, PubMed: 34278297

Bull, L., & O'Hara, T. (2002). Accuracy based neuro and neuro-fuzzy classifier systems. In *Proceedings of the 4th annual conference on Genetic and Evolutionary Computation* (pp. 905–911).

Herm, L. V., Heinrich, K., Wanner, J., & Janiesch, C. (2022). Stop ordering machine learning algorithms by their explainability! A user-centered investigation of performance and explainability. *International Journal of Information Management*, *69*, 102538. https://doi.org/10.1016/j.ijinfomgt.2022.102538

Iqbal, M., Browne, W. N., & Zhang, M. (2014). Reusing building blocks of extracted knowledge to solve complex, large-scale Boolean problems. *IEEE Transactions on Evolutionary Computation*, *18*(4), 465–480. https://doi.org/10.1109/TEVC.2013.2281537

Lanzi, P. L., & Loiacono, D. (2006). XCSF with neural prediction. In *2006 IEEE international conference on Evolutionary Computation* (pp. 2270–2276). IEEE. https://doi.org/10.1109/CEC.2006.1688588

Liu, Y., Browne, W. N., & Xue, B. (2019). Absumption to complement subsumption in learning classifier systems. In *Proceedings of the Genetic and Evolutionary Computation Conference* (pp. 410–418). Association for Computing Machinery. https://doi.org/10.1145/3321707.3321719

Liu, Y., Browne, W. N., & Xue, B. (2021a). A comparison of learning classifier systems' rule compaction algorithms for knowledge visualization. *ACM Transactions on Evolutionary Learning and Optimization*, *1*(3), Article 10. https://doi.org/10.1145/3468166

Liu, Y., Browne, W. N., & Xue, B. (2021b). Visualizations for rule-based machine learning. *Natural Computing*, *21*, 243–264. https://doi.org/10.1007/s11047-020-09840-0

Lughofer, E., Zavoianu, C., Pollak, R., Pratama, M., Meyer-Heye, P., Zörrer, H., Eitzinger, C., & Radauer, T. (2019). Autonomous supervision and optimization of product quality in a multi-stage manufacturing process based on self-adaptive prediction models. *Journal of Process Control*, *76*, 27–45. https://doi.org/10.1016/j.jprocont.2019.02.005

Margraf, A., Stein, A., Engstler, L., Geinitz, S., & Hahner, J. (2017). An evolutionary learning approach to self-configuring image pipelines in the context of carbon fiber fault detection. In *2017 16th IEEE international conference on Machine Learning and Applications (ICMLA)* (pp. 147–154). IEEE. https://doi.org/10.1109/ICMLA.2017.0-165

Permin, E., Bertelsmeier, F., Blum, M., Bützler, J., Haag, S., Kuz, S., Özdemir, D., Stemmler, S., Thombansen, U., Schmitt, R., Brecher, C., Schlick, C., Abel, D., Poprawe, R., Loosen, P., Schulz, W., & Schuh, G. (2016). Self-optimizing production systems. *Procedia CIRP*, *41*, 417–422. https://doi.org/10.1016/j.procir.2015.12.114

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "Why should I trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge Discovery and Data Mining* (pp. 1135–1144). Association for Computing Machinery. https://doi.org/10.1145/2939672.2939778

Schoettler, G., Nair, A., Luo, J., Bahl, S., Ojea, J. A., Solowjow, E., & Levine, S. (2020). Deep reinforcement learning for industrial insertion tasks with visual inputs and natural rewards. In *2020 IEEE/RSJ international conference on Intelligent Robots and Systems (IROS)* (pp. 5548–5555). IEEE. https://doi.org/10.1109/IROS45743.2020.9341714

Stein, A., Maier, R., & Hähner, J. (2017). Toward curious learning classifier systems: Combining XCS with active learning concepts. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion* (pp. 1349–1356). Association for Computing Machinery. https://doi.org/10.1145/3067695.3082488

Tan, J., Moore, J., & Urbanowicz, R. (2013). Rapid rule compaction strategies for global knowledge discovery in a supervised learning classifier system. In P. Liò, O. Miglino, G. Nicosia, S. Nolfi, & M. Pavone (Eds.), *ECAL 2013: The twelfth European conference on Artificial Life* (pp. 110–117). MIT Press. https://doi.org/10.7551/978-0-262-31709-2-ch017

Urbanowicz, R. J., Granizo-Mackenzie, A., & Moore, J. H. (2012). An analysis pipeline with statistical and visualization-guided knowledge discovery for Michigan-style learning classifier systems. *IEEE Computational Intelligence Magazine*, *7*(4), 35–45. https://doi.org/10.1109/MCI.2012.2215124, PubMed: 25431544

Urbanowicz, R. J., & Moore, J. H. (2009). Learning classifier systems: A complete introduction, review, and roadmap. *Journal of Artificial Evolution and Applications*, *2009*, Article 736398. https://doi.org/10.1155/2009/736398

Zhang, Y., Qian, C., Lv, J., & Liu, Y. (2017). Agent and cyber-physical system based self-organizing and self-adaptive intelligent shopfloor. *IEEE Transactions on Industrial Informatics*, *13*(2), 737–747. https://doi.org/10.1109/TII.2016.2618892