# An Overview of LCS Research from 2021 to 2022

Michael Heider
University of Augsburg
Augsburg, Germany
michael.heider@uni-a.de

David Pätzel
University of Augsburg
Augsburg, Germany
david.paetzel@uni-a.de

Alexander R. M. Wagner
University of Hohenheim
Hohenheim, Germany
a.wagner@uni-hohenheim.de

## ABSTRACT

The *International Workshop on Learning Classifier Systems* (IWLCS) is an annual workshop at the GECCO conference where new concepts and results regarding *learning classifier systems* (LCSs) are presented and discussed. One recurring part of the workshop agenda is a presentation that reviews and summarizes the advances made in the field over the last year; this is intended to provide an easy entry point to the most recent progress and achievements. The 2020 and 2021 presentations were accompanied by survey workshop papers, a practice which we hereby continue. We give an overview of all the LCS-related publications from 11 March 2021 to 10 March 2022. The 42 publications we review are grouped into six overall topics: Formal theoretic advances, new LCS architectures, LCS-based reinforcement learning, algorithmic improvements to existing LCSs, combinations of LCS and Deep Learning models and, finally, a variety of applications of LCSs.

## CCS CONCEPTS

• **Computing methodologies → Rule learning**; **Genetic algorithms**; • **General and reference** → *Surveys and overviews.*

## 1 INTRODUCTION

Since 2019, a recurring segment of the *International Workshop on Learning Classifier Systems* (IWLCS) is its organizers giving a presentation of an exhaustive overview of *learning classifier system* (LCS) research that was conducted over the past year. In 2020 and 2021 that presentation was accompanied by a survey workshop paper [36, 37], a practice which we hereby continue. Our primary goal is to contribute to a better organized research community; showcasing the most recent developments of the field helps to connect LCS researchers but also serves people new to the field as they can more quickly assess the latest achievements, current challenges as well as links to other areas.

In the next section, we shortly present the methodology we used for identifying publications to be included in our overview. The remaining sections correspond to six major topics that the 42 contributions we found could be divided into and a concluding summary. Note that these topics were updated as well for this year's iteration of the survey. We mention each publication only once, even if some fit into several sections; for each paper we choose the section it fits into best. If an equally strong case for multiple sections could be made, we choose the first appropriate section, in the order in which the sections appear.

## 2 RESEARCH METHODOLOGY

This survey is limited to contributions in English with *publication dates* on or after 11 March 2021 (one day after the end of the period of the previous survey) and on or before 10 March 2022. Note that some conferences may have published their proceedings within this interval despite in fact taking place at another date; since we only consider publication dates, these entries are covered by this survey as well. We intentionally *do not include military applications* of LCSs (of which we actually found one this year). Also, we leave out papers published on *arXiv* unless they are relevant for a follow-up paper which meets our criteria. Finally, we exclude publications that were already cited in the previous year's survey [36] even if they were eligible for the present survey as well (this may be the case, for example, if they have been re-published or moved from a 'pre-proof' to a 'published' status).

Our primary search tool was *Google Scholar*[1] with its time range feature as it seemed to consistently report more relevant publications than comparable tools, especially for more general search queries. We ultimately only used five different queries, two very general ones and one for each of the three major LCSs that we knew have been investigated in recent years; more specific terms did not yield relevant results that were not found by these as well (at least not in March 2022). The following lists each query with the number of Google Scholar result pages that we examined for papers meeting our requirements (numerator) and the total number of result pages (denominator); each Google Scholar result page displayed 10 publications and we set the time range filter to *Since 2021.*

- `learning "classifier system"` (44/>100)[2]
- `"evolutionary rule-based" learning` (5/5)
- `xcs classifier system` (19/19)
- `"extracs" classifier system` (1/1)

[1] https://scholar.google.com

[2] We stopped when we noticed that the results of at least ten consecutive pages did not contain any LCS-related publications that had not yet been listed on the preceding pages (i. e. we estimated the probability of further relevant results to be very low).

- `biohel` (3/3)

We also checked a number of LCS researchers (especially, but not exclusively, ones we know to be active) and the proceedings of the three main venues that have attracted LCS researchers in the past: CEC, GECCO and *Evostar*. As we did not find any publications this way that we had not already found using Google Scholar, we are confident that our Google Scholar search was sufficiently exhaustive and there was no need to further refine the used search terms.

## 3 FORMAL THEORY

There have been two spin-off papers to the work of Nakata et al. which included a proof of—if configured correctly—XCS being able to correctly *distinguish* maximally accurate rules from other rules [29] on binary-input single-step tasks with a binary rewards scheme (with correct hyperparameter settings being derived from their analysis).

The first one is by Horiuchi and Nakata [16] and extends the theory to discrete but *non-binary* reward schemes with a finite number of non-negative reward levels. The authors show that XCS is able to distinguish maximally accurate rules from other rules in this setting under similar assumptions as the ones made for the binary reward scheme (e. g. certain problem-dependent values for hyperparameters such as the learning rate are used, certain learning parameter expected values are estimated correctly) with the addition that a good estimate for a certain other parameter is available (i. e. a problem-dependent threshold $\theta_n$; how to perform this estimation in practice is deferred to future work). Experimental validation is performed using two experiments that cover the ideal case where the reward received is always the expected reward and a more practical case where the reward is sampled from the set of possible rewards. The latter experiment shows that even if the assumptions are violated, which to some degree will be the case in most practical applications, as long as the derived hyperparameter settings are used, learning performance is still better than when using the typical hyperparameter settings from the literature.

The second follow-up paper to the learning optimality theory of Nakata and Browne [29] was written by Nakamura et al. [28] who move forward formal theory regarding XCS on binary classification domains. They show that (under simplifying assumptions) XCS's rule discovery process is capable of *producing* rules with optimal generality by proving that the population's average rule generality converges to an optimal value if configured correctly. This is essentially a precondition for the learning optimality theory to work in practice. From their analysis, the authors are able to derive a rule deletion scheme that ensures convergence as well as additional guidelines for optimally setting the population size, mutation probability and GA invocation rate hyperparameters of XCS. A restriction of the conducted formal analysis is that only the average rule generality of the population as a whole is considered while the authors' simplifying assumptions include that all rules have the same optimal rule generality and that the probabilities of rules matching an input are the same. Since these assumptions are seldomly fulfilled in practice (one reason for even employing LCSs is that different parts of the input space have to be covered by differently many rules which violates these assumptions), the authors perform an experimental verification on the 70-bit multiplexer (a binary classification task) as well as the 3×3 concatenated multiplexer (a multiclass classification task). They find that their rule deletion scheme and the hyperparameter recommendations do improve performance despite the assumptions not being fulfilled on these tasks.

## 4 NEW LCS ARCHITECTURES

In their very nature LCSs are a family of learning algorithms that produce models of a shared architecture. Whereas many researchers focus on proposing advances to existing algorithms, most importantly XCS and its derivatives, some researchers still find novel approaches to train LCS models. In this section, we recount the key ideas of two innovative new algorithms.

Bishop et al. [2] present a new Reinforcement Learning (RL) technique for scenarios with continuous inputs and discrete actions. The method is a Pittsburgh-style system that uses coevolution to optimize two populations: A population of ordered sets of fuzzy logic membership functions (i. e. one fuzzy partitioning for each input feature) and a population of rule sets that use these partitionings. Combining one of the rule sets with one of the fuzzy logic membership function sets yields an RL policy which can be evaluated on the considered RL task. The system also performs multi-objective optimization with respect to the return achieved by the explored policies and the policies' complexity. The authors evaluate their approach on the well-known Mountain Car problem and also shortly discuss interpretability of the resulting rule sets.

Guendouzi and Boukra [11] propose EDE-FRMiner a differential evolution–based rule induction algorithm to generate fuzzy rules for supervised classification tasks. The matching function of a rule is based on attribute-wise triangular membership functions and rules incorporate a heuristic estimate of correctness of their class assignment. A modified version of differential evolution then serves as an optimizer within a Michigan-style approach to generate a well-suited rule set. They compare their system with other rule-based machine learning (ML) algorithms, most notably GAssist, on a variety of publicly available datasets. While EDE-FRMiner was on average the best ranked algorithm in terms of run time and achieved accuracy it produced less compact solutions than GAssist. The authors also analysed their algorithm's scalability on high-dimensional data and the noise sensitivity on an artificially created dataset, finding promising results.

## 5 REINFORCEMENT LEARNING

LCSs were originally proposed as RL algorithms and ever since this subfield of ML remains of interest to the community. In this section we introduce last year's improvements and benchmarks of existing algorithms, many of which were evaluated on game environments.

A novel modification of the ACS2 system called *Averaged ACS2* (AACS2) is proposed by Kozłowski and Unold [21]. It is based on the notion that many real-world sequential decision problems can be solved by maximizing the *average of successive rewards* (unlike the common formal notion of maximizing the return, i. e. the sum of rewards ever received or the sum of rewards within an episode). The authors explore two variants of AACS2 that differ only in the way the average reward is estimated and compare those algorithms to Q-learning, the original ACS2 as well as R-learning (another average reward–based learner) on three delayed-reward RL tasks (corridor, finite state world 20 scenarios and Woods1).

In their extended abstract, Orhand et al. [35] discuss future research directions for Anticipatory Learning Classifier Systems with respect to the extensions that were introduced by the same authors in recent years, behavioural sequences and probability-enhanced predictions. They shortly elaborate on strengths and weaknesses of the approaches and conclude that a combination of the two may prove fruitful.

Siddique et al. [45] extend their earlier work on learning at different levels of abstraction to RL scenarios (deterministic discrete 2D maze environments with aliasing states) where it can contribute to solving problems of perceptual aliasing. Their approach builds policies by combining so-called code paths (shallow trees of state-action-state sequences) and allows for the agent to utilize different levels of abstraction by choosing tree depths freely (up to a maximum depth). Aliased states are detected based on the knowledge encoded in a policy's code paths; however, this does not work in all cases and as a fallback, the agent's relative coordinates to states visited earlier are computed (this is possible due to actions moving the agent by a noiseless discrete amount in one of four directions and the chosen actions being logged). The authors conclude that their approach compares favourably to several other methods for these kinds of learning tasks including ACS2 and a deep recurrent Q-network.

RL also plays a role in Siddique's PhD thesis which he published last year [44]. It provides a detailed account of the general topic of *lateralized learning* and how it can be applied to different kinds of problems—among them, RL tasks as well as classification tasks.

Hansmeier and Platzner [14] evaluate four existing explore/exploit strategies for XCS on three RL tasks (the 11- and 20-bit multiplexers and the Maze4 environment). One of the four strategies, which are chosen based on a principled literature review, is input local (i. e. the decision for/against exploitation depends on the current RL state) whereas the other three act globally (i. e. decisions do not depend on the current RL state but rather on global population metrics). At that, all investigated strategies only perform a binary decision between pure exploration (selection of a random action) and pure exploitation (greedy action selection), that is, they do only determine *when* exploration should be performed and not *how*. The authors' parameter study shows that the strategies' hyperparameters are rather sensitive with respect to the learning task considered and that, for the scenarios they analyze, an argument can be made for at least two of the four strategies.

In [34], Oberoi et al. preliminarily investigate the feasiblity of XCS in an artificial player of a 6×6 variant of Checkers, a strategic, combinatorial game with high branching factor and complex state space. To do so, they adapted the XCS agent to the specific Checkers variant and trained it using a random agent. In the evaluation experiments in which the customized XCS agent competed against the alpha-beta pruning algorithm of different depths as well as against human agents with different capabilities, the XCS agent performed well.

Büttner and von Mammen [5] use XCS-RC, an XCS derivative with inductive reasoning rather than stochastic optimization, to play games of competitive snake. They focus on self-play-based RL, where two agents with the same model play against each other. While the agents were able to successfully navigate the game and complete objectives it became clear that some look-ahead capability to carefully plan out the next moves could have improved their overall performance.

To solve competitive Markov games (i. e. games against other players that typically show sophisticated behaviour themselves), Chen et al. [6] proposed HAMXCS (heuristic accelerated Markov games with XCS): A neural network learns to predict an opponents next move by approximating their policy. This information is then used to choose the action from the matching classifiers. Contrary to standard XCS, not only the classifiers in the previous action set but all the classifiers that match the state-action pairs are updated. HAMXCS was evaluated on the Hexcer game as well as a modified version of the thief-hunter problem and compared with several Q-learning as well as neural network-based RL systems. The results indicate that HAMXCS outperforms the baseline systems in terms of win rate; the authors acknowledge, however, that HAMXCS took significantly more computation time to train which they deem acceptable given the increased performance and the fact that the system's actions are more interpretable (which they show by performing an exemplary interpretability analysis of a selection of rules).

Novak and Fister [33] propose to perform automated software testing using XCS. They test whether a game successfully implements all its features by letting XCS learn to play the game. For validation of their approach, they use a simple game of tic tac toe where they introduced 8 exploitable bugs in the opponent's strategy and conclude that XCS is able to find and exploit all of them.

## 6 IMPROVEMENTS TO EXISTING ALGORITHMS

The XCS classifier system is the most investigated LCS to date. Over the more than 25 years of its history, a large variety of extensions and improvements to its structure and components have been proposed. One such extension to tailor it for the use in classification was the sUpervised Classifier Systems (UCS). In the following, we present improvements to XCS or UCS that were proposed during the last year.

### 6.1 Improvements and extensions for XCS

Nguyen et al. [32] define a new fitness function for code fragments (CFs) in the XOF classifier system. It includes CF complexity (measured as the number of leaf nodes in the tree) which earlier definitions of fitness were lacking and leads to generally less bloated CFs as well as makes the CF depth limit hyperparameter unnecessary. Aside from that, an accompanying niching method for generating CFs is introduced as well. For evaluation, the authors use binary classification tasks with interacting features, namely, the 11-bit even-parity problem, the 18-bit hierarchical multiplexer and the 18-bit hierarchical majority-on problem. They measure the structural efficiency of CFs used by classifiers in the population by tracking rule generality and complexity and conclude that growth of CF depth decreases and that structural efficiency of the CFs is improved only if both the fitness measure and the niching method are employed which also leads to CFs evolving more quickly without being trapped in local optima. This as well as earlier works of those authors are also part of the primary author's Nguyen's PhD thesis, which was also published last year [31].

To achieve their overarching goal, the implementation of a more reliable specialization pressure in XCS to prevent detrimental effects due to over-generalization, Wagner and Stein proposed and evaluated two different extensions for XCS for continuous valued inputs. In [48], two variants of lexicase selection, that is, batch lexicase selection and $\epsilon$ lexicase selection, are introduced for parent selection in the GA of XCSR and XCSF. Instead of an error-aggregating fitness value, lexicase selection bases the parent selection on the accuracy of the rules using test cases (i. e. previously visited environmental states) for the accuracy assessment. In addition to the adaptation of the lexicase selection variants to XCSR and XCSF, they additionaly proposed a local test case storage, the so-called classifier experience storage, which enables a niche-specific mode of operation of the lexicase selection variants. The evaluation on multiple classification tasks (e. g. the three-dimensional checkerboard with both six and eight divisions per dimension and real world data sets), and four regression tasks (e. g. the eggholder function and the sine-in-sine function), showed lexicase selection enables XCSR and XCSF to remarkably improve its learning performance, as the different problems are learned more quickly and with increased accuracy.

In [49], Wagner and Stein proposed their second extension to tackle the problem of over-generalization in XCS for continuous valued problems, the so-called *over-generality handling* (OGH). It is based on two previously introduced methods for XCS for binary inputs, namely *absumption* by Liu et al. [25] and the *specify operator* by Lanzi [23]. For the adaptation to continuously valued input spaces, two new strategies to specialize over-general rule conditions have been proposed: either smaller conditions are randomly generated, which contain the center point of the condition of the over-general rule, or smaller conditions are generated which are entirely contained inside the condition of the over-general rule. In addition, a new technique to detect over-general classifers in multi-step settings has been introduced which is based on the increase in oscillation of the prediction of an over-general rule due to matching of environmental niches with different payoff levels. The potential of OGH was evaluated using different classification tasks (e. g. Mario pixel-art and real-world data from the agricultural domain) and a multi-step problem, namely the grid world with puddles. The results showed that the application of OGH leads to considerable learning performance improvements of XCSR, especially in case of the considered real-word data and the used multi-step problem. On the other hand, OGH's underlying mode of operation leads to an increased number of transient rules during the learning phase.

The work on OGH is continued in [50]. The authors additionally fathom the potential of the application of OGH in function approximation tasks with XCSF and conduct an analysis based on three different functions (i. e. the two-dimensional sine-in-sine function, the three-dimensional cross function and the eggholder function) with different settings of the $r_0$ hyperparameter, to artificially induce different tendencies to over-generalize. The results show that OGH can also lead to significant improvements in the learning performance of XCSF: In the considered cases, it reduces the system error but at the same time, again, causes the aforementioned increase in the number of transient rules.

Preen et al. [38] propose to use LCSs for building autoencoders that consist of a set of smaller localized autoencoders (each localized autoencoder corresponding to one rule in the LCS). In the proposed system, an adapted variant of XCSF is used in which the condition part $cl.C$ and the prediction part $cl.P$ of each rule is a separate neural network. The neural network for $cl.C$ is fully connected feed forward with a single output neuron that determines whether the rule matches a given input, whereas the neural network for $cl.P$ is a small autoencoder. The number of neurons and connections in the hidden layer of the two neural networks are optimized by the GA which also adapts the weights of the $cl.C$ network; the weights of the $cl.P$ autoencoder are fitted using backpropagation. By decomposing the input space using XCSF, the small autoencoders can adapt their architecture to the local niches and thus have different complexities. Therefore, compared to a global autoencoder approach, the proposed system can be expected to lead to a reduction in convergence time, computational cost, code size and resulting decoder computational cost due to heterogeneous small autoencoders. The authors perform a study that compares their system with another version of it that does not partition the input space (i. e. $cl.C = 1$ for all inputs) to find out whether XCSF's niching has any merit in the autoencoder setting. They conclude that the niching LCS performs better in terms of convergence and reconstruction error as well as that it has more tolerance with respect to restrictions of the maximum number of neurons allowed.

In [26], Liu et al. propose a concept called *natural solution*, which describes the existence of a consistent deterministic solution for each data set containing only maximally accurate and maximally general rules. This concept extends the hypothesis about the optimal (i. e. accurate and maximally general) set of rules $[O]$ of Butz et al. [4]. In order to study the natural solution and $[O]$ along with their differences, two additional compaction algorithms are presented, namely *Razor Cluster Razor 2* (RCR2) and *Razor Cluster Razor 3* RCR3. RCR2 aims to find the natural solution, while RCR3 aims to find $[O]$. In the evaluation experiments conducted, the proposed concept of natural solutions was investigated by applying RCR2 and RCR3 to 16 artificial Boolean problems, that ranged from 6 to 70 bits, and 3 real-valued datasets. In addition, eight of the most popular previous compaction algorithms, namely CRA, FU1, FU3, CRA2, K1, QRC, PDRC, and RCR, were also reviewed in the evaluation experiments and their compaction results compared with RCR2 and RCR3. The results show that unlike the previous compaction algorithms, the proposed compaction algorithms can extract the rules that accurately contain the ground truth of a problem when the dataset is fully observable and does not contain noise. This is further evidence that LCSs are an appropriate technique for building natural, interpretable models in data-mining tasks. However, when the data set is noisy, the generated rules are more likely to reflect the ground truth of the provided dataset instead of the problem.

In their thesis [24], Liu combines their previously proposed research in the domain of LCS: the visualization techniques (such as the *Feature Importance Map* (FIM)), the natural solution [26], the new compaction algorithms (*Razor Cluster Razor* (RCR) and

*Razor Cluster Razor for real-valued domains* (RCR-Real)), the *Hierarchical Learning Classifier System* (HLCS) and the approaches to overcome the issue of over-generalization (*Absumption* and the *Absumption Subsumption based learning Classifier System* (ASCS)) as well as leveraging rule patterns across multiple interconnected populations. It is shown that LCSs are able to generate optimal solutions or natural solutions for clean datasets, that is, datasets without any noisy instances, using the introduced algorithms and techniques, and these solutions contain visible patterns that reflect the ground truth of the considered datasets. The natural solution contains all unsubsumable correct rules found in the global search space.

## 6.2 Improvements for UCS

Nazmi et al. [30] extend their UCS-based LCS for multi-label classification by introducing a covering operator that respects label dependencies. The method builds label correlation graphs and derives from them subsets of semantically close labels for each of which a rule is created. Further, a novel method for calculating prediction arrays in the multi-label setting is proposed which is based on estimating each rules' precision in predicting each individual label it contains. On the five multi-label classification tasks (PASCAL-VOC6, Scene, Mediamall, Corel5k and Corel16k; with prior dimensionality reduction to 256 features using the VGG16 model) that the authors evaluate it on, the extended system compares favourably to both the authors' earlier system (which was based on the label powerset technique) as well as several well-known multi-label classification algorithms (i. e. CLR, ECC, HOMER, ML-*k*NN, RA*k*EL).

To improve the performance of UCS on real-valued classification tasks Hamasaki and Nakata [12] propose the Minimum Rule-repair Algorithm (MRA). This algorithm attempts to slightly modulate the decision boundary so that misclassified examples are no longer matched without overly reducing rule generality. Therefore, it either removes a singular example or only changes the upper or lower value of a bound in a singular dimension. The authors evaluate MRA on a variety of real-valued multiplexer and real-valued majority on problems and find that UCS with MRA does converge considerably faster and often to better solutions (both smaller and higher performing) than standard UCS. They also find that the newly introduced hyperparameters are not very sensitive for a majority of problems.

# 7 LCS AND DEEP LEARNING

In modern ML, deep neural networks have become well established due to their strong learning capabilities. Especially the capabilities of deep convolutional networks as powerful feature extractors and dimensionality reductors motivated their use in combination with LCS. *Autoencoders* are often-favoured architectures to achieve these goals, as they can be trained unsupervisedly with unlabelled data. Whereas neural network classification is very intransparent, LCS models should allow an understanding of the classifications made based on the provided features. Therefore, several research endeavours have proposed the combination of features extracted by autoencoders (or other deep convolutional feature extractors) and LCS.

For the classification of real world images Irfan et al. [17] combine a deep autoencoder and XCSR to CAXCS. XCSR evolves interpretable rules operating on rich feature maps extracted by the encoder part of an autoencoder. These rules can then be transformed to operate on the original input features using the decoder. The authors demonstrate the applicability of their system on synsets of the ImageNet database depicting underwater scenes. They compare CAXCS with a variety of benchmark deep neural architectures for image classification on multiple combinations of two synsets each and find that the newly proposed system outperforms the selected previous architectures in terms of accuracy and F-measure. They also provide some analysis of the rules and find that around 30% of the encoded features do not contain meaningful information to classify the images.

To move towards ML systems with lifelong learning capabilities, Irfan et al. [18] propose the usage of code fragment–based LCSs in combination with deep learning feature extraction models. The neural networks are pre-trained to classify images from a domain (i. e. underwater images) and the classification layers are then replaced with an LCS using code fragments as conditions. This LCS is then trained to classify the same images using the features extracted by the networks. Upon successful training, the code fragments are stored in a knowledge base. For new tasks, such as the classification of new classes from the same domain, the feature extractor is reused and the stored code fragments can be utilized to jump start training with meaningful knowledge. The authors successfully demonstrate the applicability of this approach with a variety of typical image classification neural network architectures for feature extraction.

In [19], the same authors then combine the two previous approaches into a new system using deep autoencoder–based feature extraction and code fragment LCS–based classification. They modify mutation and subsumption and present a mechanism to generate diverse and general sets of classifiers per task and, thus, code fragments for the knowledge base. The applicability of the approach is demonstrated on three large image datasets outperforming both the baseline method and a variety of models from literature applied to similar problems in the past.

Shehu et al. [41] present a novel system based on the idea of lateralization in biological systems which performs emotion categorization in face images by considering heterogeneous features, such as mouth, eyes, nose, and jaw. Due to the application of the lateralized approach, the robustness against hostile attacks such as the one-pixel attack should be increased. To achieve this goal, the novel system performs emotion categorization as follows: At first, a face is initially detected using the *Haar cascade classifier* in a given image, which is subsequently segmented into the respective parts, namely face, jaw, eyes, mouth, and nose, using *dlib*. In the so-called *context phase*, multiple *deep neural networks* DNNs (VGG19 architecture) generate predictions either for the constituent level (i. e. the individual parts) or the holistic level (i. e. the big picture) of the given face image. In case the predictions at the constituent level and the holistic level diverge in estimation, the system is in doubt about correctly classifying the emotion of the given face image and invokes the so-called *attention phase*. In this phase, multiple *sUpervised Classifier Systems* (UCS) are used to generate either constituent-level predictions or holistic-level predictions based on

HOG features computed for segmented parts of the given face image. Finally, all predictions from both phases are analysed, and the category with the highest score is predicted. In the evaluation experiments, the novel lateralized system is shown to correctly predict severely corrupted images and therefore possesses a certain robustness against adversarial attacks. Compared to state-of-the-art deep learning models, the new lateralized system outperformed VGG19 by 15% to 36% with respect to classification accuracy. Although the novel lateralized system still performed best, it could not withstand a certain strong adversarial attack either (the classification accuracy dropped to about 51%).

To enable the detection of incorrect output of an CVAEXCSR LCS, Shiraishi et al. [43] propose in their extended abstract the *Misclassification Detection based on Conditional Variational Auto-Encoder* (MD/C). MD/C decides whether an CVAEXCSR's output $a$ for input $x$ is incorrect by applying its internal decoder to $a$ (yielding some $\hat{x}$) and then computing the difference between $x$ and $\hat{x}$. The system is evaluated using the MNIST dataset, the experimental results indicate that CVAEXCSR combined with MD/C significantly outperforms the standalone version of CVAEXCSR and of XCSR, as CVAEXCSR+MD/C achieves an accuracy of 99% compared to 88% by CVAEXCSR and about 10% by XCSR. Therefore, the new extension may improve classification performance and especially the applicablity of LCSs to high-dimensional inputs.

Shiraishi et al. [42] then build the same extension as a so-called *refinement component* into their *Encoding, Learning, Sampling, and Decoding Classifier System* (ELSDeCS) yielding the *Encoding, Learning, "Plausible" Sampling, and Decoding Classifier System* (ELPS-DeCS). To assess the impact of this change, the system is evaluated using the MNIST dataset with the unchanged ELSDeCS system as a baseline. According to the results, the refinement component, both significantly improves the accuracy and enhances the interpretability, at least in case of the MNIST dataset.

A further new variant of ELSDeCS is proposed by Tadokoro et al. in [47], namely MVN-ELSDeCS, in which the XCSR is replaced by an *XCSR based on multivariate normal distribution* (MVN-XCSR) to handle the internal representation of the variational autoencoder in ELSDeCS. In MVN-XCSR, the standard hyperrectangular rule condition representation is replaced by a multivariate normal distribution, which also results in adaptations of the matching and subsumption mechanisms of XCSR. As before, an evaluation experiment is performed on the MNIST dataset comparing MVN-ELSDeCS and standard ELSDeCS. It is found that MVN-ELSDeCS achieves significantly higher accuracy compared to ELSDeCS, as MVN-XCSR shows higher classification performance on the dimensionally compressed latent space already in the early stage of training. Furthermore, the reconstructed rules generated by MVN-ELSDeCS showed higher classification performance for the original high-dimensional data, a fact that was further exploited by additional training of an XCS on the reconstructed rules, resulting in significantly improved classification accuracy of the XCS for high-dimensional data. Overall, the use of MVN-ELSDeCS enabled the creation of rules with both high interpretability and high classification performance for high-dimensional inputs.

# 8 APPLICATIONS

An important step in the life-cycle of an algorithm is the transition from being developed and tested under laboratory conditions into real world applications and testing in the wild. In this section we want to first summarize works from the past year that present theoretical use cases and make arguments why LCS are well-suited learning algorithms in these domains. Then, we present last year's variety of applications, both on real world data sets or directly in production.

## 8.1 Theoretic applications and positions

Krupitzer et al. [22] explore a concept of hybrid system management where distributed decision making of heterogeneous autonomous agents is integrated into central optimization processes. An example use case would be a taxi service using privately-owned self-driving cars. The authors motivate the use of XCS variants for the required autonomous distributed decision making: The explainability and interpretability offered by LCSs would make them more desirable than deep learning techniques, although hybrid approaches could provide an optimal balance between explainability and interpretability and raw performance. The authors posit that due to the limited feedback, dynamic search spaces, situational action restrictions and multiple goals, which might shift during runtime, new variants of XCS would need to be developed.

Heider et al. [15] describe a scenario where humans, real-world technical systems and virtual agents collaboratively solve tasks, for example, the manufacturing of products in an industrial setting. The agent would serve as an assistance system for machine parametrizations. Based on a short literature review regarding explainability, interpretability and transparency capabilities of LCSs, they motivate the application of LCSs as the decision making component of an agent within such socio-technical systems as it is likely that these capabilities would increase human trust in the agent's decision making. The authors then formulate a template, consisting of seven direct questions, to situationally assess explainability requirements as well as LCS model design requirements.

In a short position paper [13], Hansmeier describes his plans for his PhD project which is about employing LCSs to enable self-awareness capabilities in heterogeneous compute nodes. Included in the paper is a preliminary investigation of XCS's performance on the task-to-resource assignment problem (i. e. for a computing task, deciding whether it should be run on the CPU, GPU or FPGA) which showcases that XCS is able to exploit a simple pattern in the arriving tasks. Aside from that the author shortly discusses some advantages and limitations of using XCS for this problem.

To realize *self-learning and self-organizing* (SASO) systems, machine learning techniques capable of operating in environments with dynamically changing conditions or unpredictable operational events are needed. In [46], Stein and Tomforde discuss the application of XCS or XCS-based systems in SASO systems, due to advantages such as improved interpretability and capability to continually evolve their knowledge as well as their previous successful application to self-adaptation tasks concerning condition-aware reconfiguration of parameters of productive systems. They present a system model for XCS-based SASO agents and planned improvements to the XCS algorithm to augment XCS with proactive learning behavior based on self-awareness and self-reflection capabilities, as they emphasize the importance of integrating concepts of self-reflection, flexibility, and transferability of knowledge to realize XCS-based SASO agents. It is expected that these adaptations would lead to more robust and efficient learning behaviour of the SASO agents as well as a more efficient use of existing knowledge within the overall collective system structure.

## 8.2 Real world applications

Rosenbauer et al. [40] investigate the effect of a simple population transformation that enables transfer learning for XCSF. The transformation essentially only resets rule fitness, the expected rule error estimates as well as some of the bookkeeping parameters while assuming that the dimensionality of the input space stays the same. In the considered scenario (which is, prioritizing software tests for test selection) the authors are able to show that the transfer learning regime has some albeit small benefits with respect to performance.

In a subsequent study [39], the authors combine their previous work on LCS-based test selection (including the transfer learning approach) to form a software architecture for an autonomous agent for automated test selection, building on concepts from the field of Organic Computing and evaluating the agent with respect to several metrics from that field (e. g. autonomy, self-organization and robustness).

Kato and Sbicca [20] use LCSs to investigate trust in groups of autonomous agents (one LCS instance per agent, 60 agents in total) in a grid world hunters/gatherers scenario which is intentionally designed to be close to the well-known investment game. The authors hand-craft starting rules for each of the decisions that the agents have to make and then investigate different combinations of agents being static or learning, being allowed to interact with any other agent or exclusively within certain subgroups. The authors observe that, without learning, agents act rather selfishly (which coincides with the prediction from classical game theory) whereas adding rather simple learning capabilities already leads to an emergence of trust.

Malhotra and Khanna [27] benchmark 14 algorithms (among them, GAssist, MPLCS, UCS, XCS—as they are implemented in the KEEL tool[3]) against one another regarding their performance on the task of learning to recognize change-prone software modules. The motivation behind this is that software projects can benefit from identifying such modules early on as that enables precautionary design decisions that may lead to an easier to maintain software product in the end. As predictors, for a certain software module, the well-known metrics for object-oriented software by Chidamber and Kemerer are used as well as the module's number of lines of code. The target is binary and corresponds to whether the module will be changed at least once after the software was deployed or distributed. The authors investigate 14 software projects (all Java or C++) and conclude, based on Friedman and Nemenyi tests, that LCSs at least perform well on this task if not outperform most of the other algorithms.

To predict depression from a variety of sociological factors, Billah et al. [1] first generated a dataset from a group of students and then analysed this dataset using a multi-layer perceptron, the fuzzy unordered rule induction algorithm (FURIA) and an evolutionary multi-objective rule learning system finding comparable results.

Bu et al. [3] detect database intrusion attacks when using role-based access control. They utilize an LCS to determine good input features for a deep learning model and later on construct ensembles of these individual models for the purpose of prediction. A rule is therefore determined by a binary condition (representing the individual input features) and a deep classification model with a fitness dependent on the accuracy of the model's prediction during training. To gain an architectural variety of deep classifiers in the pool of models available for the ensemble prediction this process is repeated for each architecture. The ensemble is then constructed to contain accurate but diverse classifiers. They find that this ensemble outperforms previous benchmarks of a singular model for which the input features where determined by an LCS.

Chi and Hsiao [8] analyse the biomedical reaction of 32 students to gameplay videos of video games to predict their individual risk of gaming disorder. The ground truths were determined using state of the art psychological questionnaires. They utilize XCSR with center-spread conditions and replace the action with the respective class for this supervised learning task. To determine which features were useful to make a prediction the feature selection rate was computed based on high performing classifiers across 30 runs. XCSR achieved an average training data accuracy of 90% with very high-frequency pulse rate variability as the most prominent biomarker among rules.

In a related study Chi et al. [7] aimed at predicting the risk of internet addiction in conjunction with existing internet gaming disorder among 50 student participants. Similarly, the internet gaming disorder classification and the risk for internet addiction were determined using a questionnaire and biomedical responses were measured during the consumption of gameplay videos. XCSR was then used to predict internet addiction based on the questionnaire answers and biomedical responses to stimuli in separate experiments. The average 10-fold accuracy was slightly over 75% and using the commonly deemed relevant features—by analysing classifiers matching functions—the questionnaire could be reduced from 26 to 19 questions and the biomedical data could also be limited to fewer sensors.

To generate interpretable rules for image classification De Falco et al. [9] present an approach using a differential evolution–based rule learning system. The image data is pre-processed into a set of 64 features using a static filtering technique. Subsequently, rules are induced by optimizing their conditions with a performance-based fitness function. The combination of preprocessing and rule learning is tested on a set of COVID-19 chest X-rays with high class imbalance. Given the gray-scale nature of X-rays only 29 of the 64 features are used for this study. The authors compare the performance of their classification system with various other ML techniques using the same features. While they find that their algorithm does not outperform all of the other techniques in terms of Matthews correlation, they stress that it does provide models that are much easier to interpret.

[3] http://www.keel.es/

Gowri et al. [10] propose the use of XCS to balance resource allocation for compute tasks between cloud and fog devices. They present a short scenario with inputs being workload, battery capacity and network congestion and the action being the workload to be processed in fog. The payoff signal is based on time to solution (delay) and energy usage.

## 9 SUMMARY AND CONCLUSIONS

This paper gave an overview of all the LCS-related publications since 11 March 2021, that is, since the submission of the previous such survey to the IWLCS 2021. We clustered the contributions we found into six overall topics.

Formal theory saw two contributions in the last year that were based on and advance work from the previous year. We encourage researchers to contribute further formal theory in order to, in the long term, develop more formally backed improvements to LCSs.

Two new LCS algorithmic architectures were proposed. A Pittsburgh-style reinforcement learning (RL) approach and a differential evolution-based Michigan-style system for supervised learning. Interestingly, despite being developed independently, both systems utilized fuzziness in their rules.

Just like in recent years, RL saw some community interest. We found 8 papers concerning improvements or applications to RL tasks, most notably for the ACS2 and XCS algorithms. As is not uncommon for the RL community, many test-beds were related to games.

XCS and its extension UCS saw 9 further studies (7 and 2 respectively) that proposed a wide variety of features to improve these systems. In particular, we want to highlight the works regarding explainability and transparency of LCS models.

A total of seven papers proposed the use of different deep convolutional models, most notably autoencoders, for feature extraction and LCSs as classification algorithms. The LCSs were typically motivated for their easily interpretable rules.

The successful application of techniques (and their subsequent improvements) of a field is very important to validate developments and to uncover subsequent research topics. This year we found 4 papers that raise new research topics to make algorithms fit for use in the presented use cases and 10 papers that attempted utilizing LCS in real world applications or on real world data sets.

Overall, past year's LCS research features a healthy diversity from theory over incremental algorithmic and methodical improvements to applications in new domains.

## REFERENCES

[1] Md. Al-Mamun Billah, M. Raihan, Nasif Alvi, Tamanna Akter, and Nusrat Jahan Bristy. 2021. A Data Mining Approach to Identify the Stress Level Based on Different Activities of Human. In *2021 International Conference on Information and Communication Technology for Sustainable Development (ICICT4SD)*. 31–34. https://doi.org/10.1109/ICICT4SD50815.2021.9396896

[2] Jordan T. Bishop, Marcus Gallagher, and Will N. Browne. 2021. *A Genetic Fuzzy System for Interpretable and Parsimonious Reinforcement Learning Policies*. Association for Computing Machinery, New York, NY, USA, 1630–1638. https://doi.org/10.1145/3449726.3463198

[3] Seok-Jun Bu, Han-Bit Kang, and Sung-Bae Cho. 2022. Ensemble of Deep Convolutional Learning Classifier System Based on Genetic Algorithm for Database Intrusion Detection. *Electronics* 11, 5 (2022). https://doi.org/10.3390/electronics11050745

[4] Martin V. Butz, Tim Kovacs, Pier Luca Lanzi, and Stewart W. Wilson. 2001. How XCS Evolves Accurate Classifiers. In *Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation* (San Francisco, California) *(GECCO'01)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 927–934.

[5] Johannes Büttner and Sebastian von Mammen. 2021. Training a Reinforcement Learning Agent based on XCS in a Competitive Snake Environment. In *2021 IEEE Conference on Games (CoG)*. 1–5. https://doi.org/10.1109/CoG52621.2021.9619161

[6] Hao Chen, Chang Wang, Jian Huang, and Jianxing Gong. 2021. Efficient use of heuristics for accelerating XCS-based policy learning in Markov games. *Swarm and Evolutionary Computation* 65 (2021), 100914. https://doi.org/10.1016/j.swevo.2021.100914

[7] Hung-Ming Chi, Liang-Yu Chen, and Tzu-Chien Hsiao. 2021. Extraction of Psychological Symptoms and Instantaneous Respiratory Frequency as Indicators of Internet Addiction Using Rule-Based Machine Learning. *Advances in Science, Technology and Engineering Systems Journal* 6, 5 (2021), 203–212.

[8] Hung-Ming Chi and Tzu-Chien Hsiao. 2021. Extended classifier system with continuous real-coded variables for feature extraction of instantaneous pulse-rate variability and respiration of individuals with gaming disorder. *BioMedical Engineering OnLine* 20, 1 (23 Sep 2021), 93. https://doi.org/10.1186/s12938-021-00930-3

[9] Ivanoe De Falco, Giuseppe De Pietro, and Giovanna Sannino. 2022. Classification of Covid-19 chest X-ray images by means of an interpretable evolutionary rule-based approach. *Neural Computing and Applications* (2022), 1–11.

[10] A. S. Gowri, P. ShanthiBala, and Immanuel Zion Ramdinthara. 2022. *Fog-Cloud Enabled Internet of Things Using Extended Classifier System (XCS)*. Springer International Publishing, Cham, 163–189. https://doi.org/10.1007/978-3-030-87059-1_6

[11] Wassila Guendouzi and Abdelmadjid Boukra. 2021. A new differential evolution algorithm for cooperative fuzzy rule mining: application to anomaly detection. *Evolutionary Intelligence* (2021), 1–12.

[12] Koki Hamasaki and Masaya Nakata. 2022. Minimum Rule-Repair Algorithm for Supervised Learning Classifier Systems on Real-Valued Classification Tasks. In *Metaheuristics and Nature Inspired Computing*, Bernabé Dorronsoro, Farouk Yalaoui, El-Ghazali Talbi, and Grégoire Danoy (Eds.). Springer International Publishing, Cham, 137–151.

[13] Tim Hansmeier. 2021. Self-Aware Operation of Heterogeneous Compute Nodes Using the Learning Classifier System XCS. In *Proceedings of the 11th International Symposium on Highly Efficient Accelerators and Reconfigurable Technologies* (Online, Germany) *(HEART '21)*. Association for Computing Machinery, New York, NY, USA, Article 11, 2 pages. https://doi.org/10.1145/3468044.3468055

[14] Tim Hansmeier and Marco Platzner. 2021. *An Experimental Comparison of Explore/Exploit Strategies for the Learning Classifier System XCS*. Association for Computing Machinery, New York, NY, USA, 1639–1647. https://doi.org/10.1145/3449726.3463159

[15] Michael Heider, Richard Nordsieck, and Jörg Hähner. 2021. Learning classifier systems for self-explaining socio-technical-systems. In *LIFELIKE 2020 & LIFELIKE 2021, Lifelike Computing Systems Workshop 2020 and 2021: Joint Proceedings of the LIFELIKE 2020 - 8th Edition in the Evolution of the Workshop Series of Autonomously Learning and Optimizing Systems (SAOS) co-located with 2020 Conference on Artificial Life (ALIFE 2020), Online, July 16th, 2020, and the LIFELIKE 2021 - 9th Edition in the Evolution of the Workshop Series of Autonomously Learning and Optimizing Systems (SAOS) co-located with 2021 Conference on Artificial Life (ALIFE 2021), Online, July 19th, 2021*, Anthony Stein, Sven Tomforde, Jean Botev, and Peter Lewis (Eds.). http://ceur-ws.org/Vol-3007/2021-short-3.pdf

[16] Motoki Horiuchi and Masaya Nakata. 2021. Extended Learning Optimality Theory for the XCS classifier system on Multiple Reward Scheme. In *2021 IEEE Symposium Series on Computational Intelligence (SSCI)*. 1–8. https://doi.org/10.1109/SSCI50451.2021.9660180

[17] Muhammad Irfan, Zheng Jiangbin, Muhammad Iqbal, and Muhammad Hassan Arif. 2021. Enhancing learning classifier systems through convolutional autoencoder to classify underwater images. *Soft Computing* 25, 15 (01 Aug 2021), 10423–10440. https://doi.org/10.1007/s00500-021-05738-w

[18] Muhammad Irfan, Zheng Jiangbin, Muhammad Iqbal, and Muhammad Hassan Arif. 2021. A novel lifelong learning model based on cross domain knowledge extraction and transfer to classify underwater images. *Information Sciences* 552 (2021), 80–101. https://doi.org/10.1016/j.ins.2020.11.048

[19] Muhammad Irfan, Zheng Jiangbin, Muhammad Iqbal, Zafar Masood, and Muhammad Hassan Arif. 2022. Knowledge extraction and retention based continual learning by using convolutional autoencoder-based learning classifier system. *Information Sciences* 591 (2022), 287–305. https://doi.org/10.1016/j.ins.2022.01.043

[20] Jefferson Satoshi Kato and Adriana Sbicca. 2021. Bounded Rationality, Group Formation and the Emergence of Trust: An Agent-Based Economic Model. *Computational Economics* (11 Nov 2021). https://doi.org/10.1007/s10614-021-10158-x

[21] Norbert Kozłowski and Olgierd Unold. 2021. Anticipatory Classifier System with Average Reward Criterion in Discretized Multi-Step Environments. *Applied Sciences* 11, 3 (2021). https://doi.org/10.3390/app11031098

[22] Christian Krupitzer, Christian Gruhl, Bernhard Sick, and Sven Tomforde. 2022. Proactive hybrid learning and optimisation in self-adaptive systems: The swarm-fleet infrastructure scenario. *Information and Software Technology* 145 (2022), 106826. https://doi.org/10.1016/j.infsof.2022.106826

[23] Pier Luca Lanzi. 1997. A Study on the Generalization Capabilities of XCS. In *Proceedings of the Seventh International Conference on Genetic Algorithms*. 418–425.

[24] Yi Liu. 2021. *Learning Classifier Systems for Understanding Patterns in Data*. Ph. D. Dissertation. https://doi.org/10.26686/wgtn.14870208.v1

[25] Yi Liu, Will N. Browne, and Bing Xue. 2019. Absumption to Complement Subsumption in Learning Classifier Systems. In *Proceedings of the Genetic and Evolutionary Computation Conference* (Prague, Czech Republic) *(GECCO '19)*. Association for Computing Machinery, New York, NY, USA, 410–418. https://doi.org/10.1145/3321707.3321719

[26] Yi Liu, Will N. Browne, and Bing Xue. 2021. A Comparison of Learning Classifier Systems' Rule Compaction Algorithms for Knowledge Visualization. *ACM Trans. Evol. Learn. Optim.* 1, 3, Article 10 (aug 2021), 38 pages. https://doi.org/10.1145/3468166

[27] Ruchika Malhotra and Megha Khanna. 2021. On the applicability of search-based algorithms for software change prediction. *International Journal of System Assurance Engineering and Management* (17 Apr 2021). https://doi.org/10.1007/s13198-021-01099-7

[28] Yoshiki Nakamura, Motoki Horiuchi, and Masaya Nakata. 2021. Convergence Analysis of Rule-Generality on the XCS Classifier System. In *Proceedings of the Genetic and Evolutionary Computation Conference* (Lille, France) *(GECCO '21)*. Association for Computing Machinery, New York, NY, USA, 332–339. https://doi.org/10.1145/3449639.3459274

[29] Masaya Nakata and Will N. Browne. 2021. Learning Optimality Theory for Accuracy-Based Learning Classifier Systems. *IEEE Transactions on Evolutionary Computation* 25, 1 (2021), 61–74. https://doi.org/10.1109/TEVC.2020.2994314

[30] Shabnam Nazmi, Abdollah Homaifar, and Mohd Anwar. 2021. An Effective Action Covering for Multi-Label Learning Classifier Systems: A Graph-Theoretic Approach. In *Proceedings of the Genetic and Evolutionary Computation Conference* (Lille, France) *(GECCO '21)*. Association for Computing Machinery, New York, NY, USA, 340–348. https://doi.org/10.1145/3449639.3459372

[31] Trung Nguyen. 2021. *Pattern and Function Transfer for Learning Classifier Systems*. Ph. D. Dissertation. https://doi.org/10.26686/wgtn.14411120.v1

[32] Trung B. Nguyen, Will N. Browne, and Mengjie Zhang. 2021. Constructing Complexity-efficient Features in XCS with Tree-based Rule Conditions. In *2021 IEEE Congress on Evolutionary Computation (CEC)*. 296–303. https://doi.org/10.1109/CEC45853.2021.9504784

[33] Damijan Novak and Iztok Fister. 2021. Game Feature Validation of a Real-Time Game Space with an eXtended Classifier System. In *2021 IEEE 25th International Conference on Intelligent Engineering Systems (INES)*. 000095–000102. https://doi.org/10.1109/INES52918.2021.9512921

[34] Karmanya Oberoi, Sarthak Tandon, Abhishek Das, and Swati Aggarwal. 2021. An Evolutionary Approach to Combinatorial Gameplaying Using Extended Classifier Systems. In *Applications of Artificial Intelligence and Machine Learning*, Ankur Choudhary, Arun Prakash Agrawal, Rajasvaran Logeswaran, and Bhuvan Unhelkar (Eds.). Springer Singapore, Singapore, 723–738.

[35] Romain Orhand, Anne Jeannin-Girardon, Pierre Parrend, and Pierre Collet. 2021. *Explainability and Performance of Anticipatory Learning Classifier Systems in Non-Deterministic Environments*. Association for Computing Machinery, New York, NY, USA, 163–164. https://doi.org/10.1145/3449726.3459510

[36] David Pätzel, Michael Heider, and Alexander R. M. Wagner. 2021. *An Overview of LCS Research from 2020 to 2021*. Association for Computing Machinery, New York, NY, USA, 1648–1656. https://doi.org/10.1145/3449726.3463173

[37] David Pätzel, Anthony Stein, and Masaya Nakata. 2020. An overview of LCS research from IWLCS 2019 to 2020. In *Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion (GECCO '20), July 2020, Cancun, Mexico*, Carlos Artemio Coello Coello (Ed.). https://doi.org/10.1145/3377929.3398105

[38] Richard J. Preen, Stewart W. Wilson, and Larry Bull. 2021. Autoencoding With a Classifier System. *IEEE Transactions on Evolutionary Computation* 25, 6 (2021), 1079–1090. https://doi.org/10.1109/TEVC.2021.3079320

[39] Lukas Rosenbauer, David Pätzel, Anthony Stein, and Jörg Hähner. 2021. An Organic Computing System for Automated Testing. In *Architecture of Computing Systems*, Christian Hochberger, Lars Bauer, and Thilo Pionteck (Eds.). Springer International Publishing, Cham, 135–149.

[40] Lukas Rosenbauer, David Pätzel, Anthony Stein, and Jörg Hähner. 2021. Transfer Learning for Automated Test Case Prioritization Using XCSF. In *Applications of Evolutionary Computation*, Pedro A. Castillo and Juan Luis Jiménez Laredo (Eds.). Springer International Publishing, Cham, 681–696.

[41] Harisu Abdullahi Shehu, Abubakar Siddique, Will N. Browne, and Hedwig Eisenbarth. 2021. Lateralized Approach for Robustness Against Attacks in Emotion Categorization from Images. In *Applications of Evolutionary Computation*, Pedro A. Castillo and Juan Luis Jiménez Laredo (Eds.). Springer International Publishing, Cham, 469–485.

[42] Hiroki Shiraishi, Masakazu Tadokoro, Yohei Hayamizu, Yukiko Fukumoto, Hiroyuki Sato, and Keiki Takadama. 2021. Increasing Accuracy and Interpretability of High-Dimensional Rules for Learning Classifier System. In *2021 IEEE Congress on Evolutionary Computation (CEC)*. 311–318. https://doi.org/10.1109/CEC45853.2021.9504733

[43] Hiroki Shiraishi, Masakazu Tadokoro, Yohei Hayamizu, Yukiko Fukumoto, Hiroyuki Sato, and Keiki Takadama. 2021. *Misclassification Detection Based on Conditional VAE for Rule Evolution in Learning Classifier System*. Association for Computing Machinery, New York, NY, USA, 169–170. https://doi.org/10.1145/3449726.3459508

[44] Abubakar Siddique. 2021. *Lateralized Learning to Solve Complex Problems*. Ph. D. Dissertation. https://doi.org/10.26686/wgtn.16655416.v1

[45] Abubakar Siddique, Will N. Browne, and Gina M. Grimshaw. 2022. Frames-of-Reference-Based Learning: Overcoming Perceptual Aliasing in Multistep Decision-Making Tasks. *IEEE Transactions on Evolutionary Computation* 26, 1 (2022), 174–187. https://doi.org/10.1109/TEVC.2021.3102241

[46] Anthony Stein and Sven Tomforde. 2021. Reflective Learning Classifier Systems for Self-Adaptive and Self-Organising Agents. In *2021 IEEE International Conference on Autonomic Computing and Self-Organizing Systems Companion (ACSOS-C)*. 139–145. https://doi.org/10.1109/ACSOS-C52956.2021.00043

[47] Masakazu Tadokoro, Hiroyuki Sato, and Keiki Takadama. 2021. XCS with Weight-based Matching in VAE Latent Space and Additional Learning of High-Dimensional Data. In *2021 IEEE Congress on Evolutionary Computation (CEC)*. 304–310. https://doi.org/10.1109/CEC45853.2021.9504909

[48] Alexander R. M. Wagner and Anthony Stein. 2021. *Adopting Lexicase Selection for Michigan-Style Learning Classifier Systems with Continuous-Valued Inputs*. Association for Computing Machinery, New York, NY, USA, 171–172. https://doi.org/10.1145/3449726.3459533

[49] Alexander R. M. Wagner and Anthony Stein. 2021. On the Effects of Absumption for XCS with Continuous-Valued Inputs. In *Applications of Evolutionary Computation*, Pedro A. Castillo and Juan Luis Jiménez Laredo (Eds.). Springer International Publishing, Cham, 697–713.

[50] Alexander R. M. Wagner and Anthony Stein. 2022. Mechanisms to Alleviate Over-Generalization in XCS for Continuous-Valued Input Spaces. *SN Computer Science* 3, 2 (28 Feb 2022), 176. https://doi.org/10.1007/s42979-022-01060-w