Learning High-Order Feature Interactions in Supervised Learning Classifier Systems Using Code Fragments

Le Ma
Wellington Institute of Technology
Wellington, New Zealand
lema3213@gmail.com

Abubakar Siddique
Auckland University of Technology
Auckland, New Zealand
abubakar.siddique@aut.ac.nz

Trung Nguyen
Auckland University of Technology
Auckland, New Zealand
trung.nguyen@aut.ac.nz

Muhammad Iqbal
Higher Colleges of Technology
Fujairah, United Arab Emirates
miqbal1@hct.ac.ae

Will N. Browne
Queensland University of Technology
Brisbane, Australia
will.browne@qut.edu.au

Abstract

Learning Classifier Systems (LCSs) are rule-based evolutionary frameworks that combine global search with local learning to produce interpretable predictive models. Supervised LCSs, such as ExSTraCS, have demonstrated strong performance on noisy and high-dimensional datasets; however, their reliance on fixed-length, interval-based rule conditions limits their ability to capture higher-order feature interactions. To address this limitation, we propose Code Fragment-based ExSTraCS (CF-ExSTraCS), a novel supervised LCS framework that integrates genetic programming–based code fragments into the ExSTraCS architecture. Experimental results demonstrate that CF-ExSTraCS substantially outperforms the baseline ExSTraCS system on high-dimensional Boolean tasks and consistently achieves superior or comparable performance across diverse visual feature representations.

CCS Concepts

• **Computing methodologies** → **Rule learning**; **Supervised learning**.

Keywords

Feature Patterns, Code Fragment, Learning Classifier Systems, rule-based machine learning, feature selection, explainable AI

ACM Reference Format:
Le Ma, Abubakar Siddique, Trung Nguyen, Muhammad Iqbal, and Will N. Browne. 2026. Learning High-Order Feature Interactions in Supervised Learning Classifier Systems Using Code Fragments. In *Genetic and Evolutionary Computation Conference (GECCO Companion '26), July 13–17, 2026, San Jose, Costa Rica.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3795101.3805345

1 Introduction

Learning Classifier Systems (LCSs) are evolutionary machine learning frameworks that combine population-based global search with rule-based knowledge representation to produce predictive yet interpretable models. They provide a flexible and adaptive learning paradigm by combining the global exploration capabilities of evolutionary algorithms with the local optimisation of supervised or reinforcement learning [1, 2, 14]. LCS-based systems are inherently interpretable, hence are well aligned with the growing demand for eXplainable Artificial Intelligence (XAI). Compared to black-box learning models, LCSs often achieve competitive performance while offering superior interpretability [4, 6, 8, 10].

Extended Supervised Tracking and Classifying System (ExSTraCS) represents a state-of-the-art supervised Learning Classifier Systems approach [15]. However, ExSTraCS primarily relies on fixed-length, interval-based attribute–value condition representations. While effective for problems of moderate dimensionality, this representation limits the system's ability to efficiently capture complex, higher-order, and reusable building blocks of knowledge (feature patterns) [11, 14].

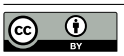
In this study, we propose Code Fragment-based ExSTraCS (CF-ExSTraCS), a novel supervised LCS framework that integrates GP-evolved code fragments into the ExSTraCS architecture. The proposed system aims to improve classification performance by uncovering informative, high-order relationships among features, while preserving the interpretability of the evolved rule set.

2 Code Fragment–Based ExSTraCS

The Extended Supervised Tracking and Classifying System is selected as the base framework for implementing strategies that evolve reusable building blocks of knowledge in supervised learning. In the proposed Code Fragment–based ExSTraCS (CF-ExSTraCS), classifier rules are redesigned to incorporate CF-based expression conditions, enabling the representation of complex nonlinear feature interactions. To support this integration, several ExSTraCS components—including classifier representation, matching, covering, genetic operators, and subsumption—are extended or modified. An overview of CF-ExSTraCS, highlighting the original ExSTraCS components that are enhanced or modified, is shown in Fig. 1.

2.1 Code Fragment–Based Condition

In CF-ExSTraCS, each rule condition is represented as a code fragment (GP tree) in prefix notation [7, 11], where terminal nodes correspond to feature variables and internal nodes represent function



GECCO Companion '26, San Jose, Costa Rica

ACM ISBN 979-8-4007-2488-6/2026/07
https://doi.org/10.1145/3795101.3805345

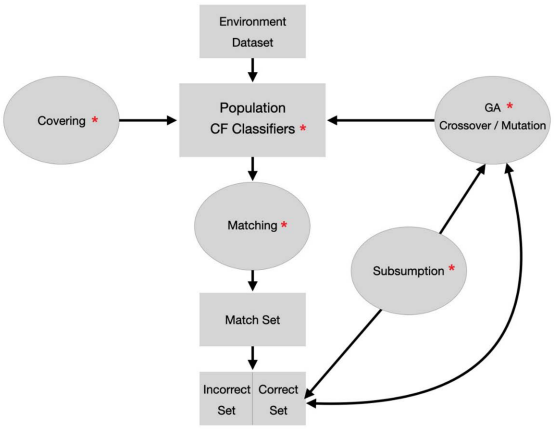


Figure 1: An overview of Code Fragment based ExSTraCS (CF-ExSTraCS), $*$ indicates the original ExSTraCS components that are enhanced or modified in this study.

operators. Two distinct function sets are defined: Boolean functions and continuous functions. The Boolean function set includes five logical operators, $\{AND, OR, NOT, NAND, NOR\}$, while the continuous function set consists of four arithmetic operators and two trigonometric functions, $\{+, -, *, /, \sin, \cos\}$, enabling numerical and nonlinear transformations. Terminal nodes are drawn from the terminal set $\{D_0, D_1, \ldots, D_{n-1}\}$, where n denotes the number of attributes in the classifier condition. Code fragments are randomly initialised by selecting internal nodes from the appropriate function set and terminal nodes from the terminal set. Tree depth is constrained with a minimum depth of one and a maximum depth of two for each CF. Additional representational power can be obtained by hierarchically stacking multiple CF levels within a condition. To mitigate the risk of overfitting or information loss, CFs are applied only to a limited subset of specified conditions, controlled by a predefined don't-care limit.

2.2 Matching Classifier

For each CF condition, matching proceeds in two stages. First, the CF expression is evaluated by substituting the relevant input feature values into the expression tree, recursively computing intermediate node outputs from the leaf terminals to the root according to the tree structure, and producing a final numerical result. The resulting scalar output is then compared against a predefined threshold. For Boolean environments, the threshold is set to 1, and the CF matches only if the output exactly equals this value. For real-valued environments, the threshold is 0.5, and the CF is considered matched if the output is greater than or equal to the threshold.

2.3 Covering

During covering in CF-ExSTraCS, a CF expression tree is randomly generated using features from the current instance and operators from the appropriate function set (see Section 2.1). The generated CF is evaluated using the matching procedure described in Section 2.2. If the CF matches the current instance, it is incorporated into the classifier; otherwise, it is discarded and regenerated. This process is repeated until CFs are created for all specified conditions of the classifier.

2.4 Genetic Algorithm

In CF-ExSTraCS, both crossover and mutation operators are designed to effectively manipulate code fragment (CF)–based rule representations while preserving their structural validity. The crossover operator follows the standard two-point strategy employed in XCSCFC [3], whereas the mutation operator is adapted from the same work to explicitly account for CF semantics. Specifically, two crossover points are randomly selected from the ordered list of specified conditions in each parent classifier, and all conditions located between these points are exchanged to produce offspring classifiers. The mutation operator consists of two complementary components: condition-level mutation and phenotype mutation. During condition-level mutation, each condition within a classifier is examined sequentially. If a randomly generated probability falls below the predefined mutation rate, the corresponding condition is selected for modification. Finally, phenotype mutation is applied independently using the same mutation rate, whereby the classifier's predicted action (phenotype) is flipped.

2.5 Subsumption

Subsumption is employed to replace classifiers that achieve the same accuracy but demonstrate inferior generalization. With the introduction of CF conditions, the comparison process is extended to evaluate CF-based conditions. A classifier can only subsume another if it is strictly more general based on CFs. The comparison proceeds in two steps: (i) Condition count. A classifier with fewer total specified conditions is considered more general. (ii) Condition set inclusion. If the condition set of one classifier is a strict subset of another, it is regarded as more general.

2.6 Overall Strategy

CF-ExSTraCS integrates CF-based conditions into the ExSTraCS framework to enhance expressiveness while maintaining interpretability. The overall learning process follows the standard UCS cycle of covering, matching, genetic search, and subsumption. At each learning iteration, the system processes an environmental instance. If no classifier matches the instance, covering is triggered to generate a new classifier through CF generation, expression evaluation, and match validation. As learning progresses, classifiers are evaluated, evolved through GA operations, and compacted via subsumption to maintain an efficient population. Classifier construction proceeds in three stages:

(1) *CF generation.* CF expression trees are randomly generated using the CF-based condition procedure described in Section 2.1.
(2) *Expression evaluation.* Feature values from the current instance are substituted into the corresponding terminal nodes of each CF, and the expression is evaluated to produce a scalar output.
(3) *Fit determination.* If the resulting value satisfies the matching criteria, the CF is incorporated into the classifier condition; otherwise, the process is repeated.

Table 1: Classification Test Accuracy for Boolean Problems
Highest Accuracy is in bold. The 6-, 11-, and 20-bit MUX experiments were conducted for 100,000 iterations with $p_{spec} = 0.67$, while the 37-bit MUX experiments were conducted for 500,000 iterations with $p_{spec} = 0.5$.

		ExSTraCS	CF-ExSTraCS
	Population	Test Accuracy	Test Accuracy
6-bits Mux	500	0.8975 ± 0.1144	**0.95 ± 0.085**
11-bits Mux	1000	0.9254 ± 0.0116	**1 ± 0**
20-bits Mux	2000	0.6657 ± 0.1711	**0.9996 ± 0.0004**
37-bits Mux	5000	0.5 ± 0.0001	**0.998 ± 0.0008**

Because the resulting classifier is tailored to the current instance, it is directly inserted into the population, Match Set, and Correct Set, and its accuracy and fitness are initialized following the standard ExSTraCS update rules. As learning progresses, each incoming instance is evaluated against the existing classifier population. For CF-based conditions, the instance's feature values are substituted into the CF expressions and compared against their corresponding thresholds. A classifier is added to the Match Set only if all of its conditions are satisfied, and it is included in the Correct Set if its predicted class label matches the true label. When the genetic algorithm is triggered, two classifiers are selected from the Correct Set to generate offspring. Crossover is performed by exchanging conditions, while mutation introduces additional variation through CF regeneration. Prior to insertion, offspring classifiers undergo subsumption, such that only classifiers exhibiting greater generality are retained.

In summary, the overall CF-ExSTraCS strategy integrates CF-based expressiveness while following the UCS cycle of covering, matching, genetic search, and subsumption. This design allows CF-ExSTraCS to balance interpretability, expressive power, and population compactness.

3 Experimental Work

To evaluate the effectiveness of the proposed CF-ExSTraCS system, a comprehensive set of experiments is conducted using both Boolean benchmark problems and computer vision datasets. For the computer vision domain, experiments are conducted using a publicly available dataset comprising three object categories: cats, dogs, and birds, referred to as the High-Resolution Cat–Dog–Bird Image Dataset [5]. The CF-ExSTraCS uses the parameter values that have been commonly used in the literature [3, 12, 15]. All the experiments have been repeated 10 times and the values presented here are obtained by averaging the results of those experiments.

Table 1 presents the classification test accuracy achieved by the baseline ExSTraCS system and the proposed CF-ExSTraCS framework on a set of multiplexer (MUX) problems of increasing dimensionality. For the lower-dimensional Boolean problems (6-bit and 11-bit MUX), both methods demonstrate strong performance; however, CF-ExSTraCS consistently achieves higher average test accuracy with reduced variability. As problem complexity increases for the 20-bit and 37-bit MUX tasks, the performance difference between the two systems becomes substantial. The baseline ExSTraCS system exhibits a marked decline in accuracy, with high variance on the 20-bit MUX problem and performance approaching random classification on the 37-bit MUX problem. In contrast, CF-ExSTraCS maintains near-perfect classification accuracy across both high-dimensional problems, with minimal standard deviation.

Table 2: Classification Test Accuracy for Cat-Dog Problems
Highest Accuracy is in bold. All experiments were conducted for 200,000 iterations with $p_{spec} = 0.67$, except for GHLS, which was evaluated using $p_{spec} = 0.33$.

		ExSTraCS	CF-ExSTraCS
	Population	Test Accuracy	Test Accuracy
GLCM	2000	0.5107 ± 0.0075	**0.579 ± 0.0067**
	5000	0.5167 ± 0.0126	**0.5839 ± 0.0076**
	10000	0.5208 ± 0.0049	**0.5805 ± 0.0073**
HOG	2000	0.5643 ± 0.0118	**0.6052 ± 0.0059**
	5000	0.5527 ± 0.015	**0.622 ± 0.0076**
	10000	0.5435 ± 0.009	**0.6109 ± 0.0098**
LBP	2000	0.5347 ± 0.0173	**0.5994 ± 0.0115**
	5000	0.5409 ± 0.0157	**0.61 ± 0.0083**
	10000	0.5385 ± 0.0102	**0.6115 ± 0.0090**
SIFT	2000	0.5364 ± 0.015	**0.5709 ± 0.0075**
	5000	0.5264 ± 0.0099	**0.5761 ± 0.0100**
	10000	0.5098 ± 0.0052	**0.5798 ± 0.0139**
GHLS	2000	**0.5853 ± 0.0131**	**0.5856 ± 0.0200**
	5000	0.5985 ± 0.0102	**0.6104 ± 0.0132**
	10000	0.5923 ± 0.0143	**0.6158 ± 0.0001**

Two sets of computer vision classification experiments are conducted: Cat–Dog classification and Cat–Bird classification. For both tasks, all images are resized to 256×256 prior to feature extraction. Four commonly used hand-crafted feature representations are used: Histogram of Oriented Gradients, Scale-Invariant Feature Transform, Grey-Level Co-occurrence Matrix, and Local Binary Patterns. Additionally, a combined feature representation (denoted GHLS) is constructed by concatenating GLCM, HOG, LBP, and SIFT features. For HOG feature extraction, pixels-per-cell values of 32 are used, while 24 GLCM features are extracted. LBP features are computed with a radius of 3, and SIFT features are extracted using 4 spatial bins. Due to the high dimensionality of HOG and SIFT features, principal component analysis (PCA) is applied to reduce feature dimensionality to 32 prior to learning. These parameter settings for feature extractions are consistent with those commonly used in prior studies [9, 13].

The classification test accuracy for the Cat–Dog problem across multiple feature representations and population sizes is presented in Table 2. For single-feature representations, CF-ExSTraCS yields substantial and consistent performance gains. Using GLCM features, CF-ExSTraCS improves test accuracy by approximately 6˜7% across all population sizes, while also exhibiting lower variance, indicating more stable learning behavior. Similar trends are observed for HOG features, where CF-ExSTraCS achieves its strongest improvement at a population size of 5000, outperforming ExSTraCS by nearly 7%.

With LBP features, CF-ExSTraCS achieves gains of approximately 6˜7% across all population sizes, along with reduced standard deviation. In the case of SIFT features, where the baseline ExSTraCS exhibits a notable performance drop at larger population sizes, CF-ExSTraCS not only mitigates this degradation but also achieves its highest accuracy at the largest population size, indicating more effective utilization of complex local descriptors.

Table 3: Classification Test Accuracy for Cat-Bird Problems
Highest Accuracy is in bold. All experiments were conducted for 200,000 iterations with $p_{spec} = 0.67$, except for GHLS, which was evaluated using $p_{spec} = 0.33$.

		ExSTraCS	CF-ExSTraCS
	Population	Test Accuracy	Test Accuracy
GLCM	2000	**0.709 ± 0.0036**	**0.7095 ± 0.0056**
	5000	0.7119 ± 0.0066	**0.7157 ± 0.0065**
	10000	**0.7133 ± 0.0037**	**0.7131 ± 0.0050**
HOG	2000	0.6896 ± 0.0102	**0.7036 ± 0.0090**
	5000	0.6996 ± 0.0106	**0.71 ± 0.0087**
	10000	0.7023 ± 0.0062	**0.7154 ± 0.0045**
LBP	2000	0.7074 ± 0.0065	**0.7387 ± 0.0077**
	5000	0.7154 ± 0.0048	**0.7474 ± 0.0098**
	10000	0.7172 ± 0.0029	**0.7522 ± 0.0052**
SIFT	2000	**0.6668 ± 0.0048**	**0.6665 ± 0.0112**
	5000	**0.6785 ± 0.0048**	0.675 ± 0.0077
	10000	**0.6862 ± 0.0094**	0.6777 ± 0.0071
GHLS	2000	**0.7743 ± 0.0096**	**0.7771 ± 0.0080**
	5000	0.7876 ± 0.0046	**0.7956 ± 0.0080**
	10000	0.7966 ± 0.0076	**0.8004 ± 0.0094**

With combined GHLS features, the performance differences between the two methods are modest at smaller population sizes, the CF-ExSTraCS demonstrates a clear advantage as population size increases. At a population size of 10,000, CF-ExSTraCS achieves the best overall result, coupled with a markedly lower variance, highlighting its ability to effectively exploit rich, multi-feature representations while maintaining stable convergence.

Table 3 presents the classification test accuracy for the Cat–Bird problem across different feature representations and population sizes. Compared to the Cat–Dog task, overall accuracy values are higher, indicating that the Cat–Bird classification problem is comparatively less ambiguous. Across most feature types and population settings, CF-ExSTraCS achieves equal or superior performance relative to the baseline ExSTraCS system.

For GLCM features, both methods exhibit very similar performance across all population sizes, with marginal differences in accuracy. CF-ExSTraCS achieves slight improvements at intermediate population sizes, while performance remains statistically comparable at smaller and larger populations. This suggests that texture-based features alone provide sufficient discriminatory information for this task, limiting the performance gap between the two systems.

More pronounced improvements are observed for HOG and LBP features. Using HOG features, CF-ExSTraCS consistently outperforms ExSTraCS across all population sizes, with accuracy gains increasing as the population size grows. A similar trend is evident for LBP features, where CF-ExSTraCS achieves substantial improvements of approximately 3˜4% across all population settings, along with consistently low variance. These results indicate that CF-ExSTraCS is particularly effective at exploiting local gradient- and pattern-based descriptors in this domain.

For SIFT features, the baseline ExSTraCS system slightly outperforms CF-ExSTraCS across all population sizes, although the performance differences are relatively small. The combined GHLS feature representation yields the highest overall classification accuracy for both systems. CF-ExSTraCS consistently achieves the best performance across all population sizes, with the highest accuracy observed at a population size of 10,000. The steady improvement in accuracy with increasing population size, coupled with competitive variance, highlights CF-ExSTraCS's ability to effectively integrate complementary information from multiple feature types.

4 Conclusion

This work introduces CF-ExSTraCS, a supervised Learning Classifier System that integrates genetic programming–based code fragments into the ExSTraCS framework to enhance rule expressiveness and learning capacity. In high-dimensional multiplexer problems, CF-ExSTraCS achieves near-perfect accuracy with minimal variance, highlighting its scalability and robustness. For computer vision tasks, CF-ExSTraCS consistently improves predictive performance across diverse feature representations, while maintaining stable learning dynamics. Future work will explore further extensions of CF-ExSTraCS, including optimization of code fragment operators, adaptive feature selection, and integration with deep feature extractors to expand applicability to real-world domains such as medical imaging and bioinformatics.

References

[1] Martin V Butz and Wolfgang Stolzmann. 2001. An algorithmic description of ACS2. In *Int. Work. Learning Classifier Systems*. Springer, 211–229.

[2] JH Holland and JS Reitman. 1998. Cognitive systems based on adaptive algorithms Reprinted in: Evolutionary computation. The fossil record. *IEEE Press, New York* (1998).

[3] Muhammad Iqbal, Will N Browne, and Mengjie Zhang. 2013. Reusing building blocks of extracted knowledge to solve complex, large-scale boolean problems. *IEEE Trans. Evol. Comput.* 18, 4 (2013), 465–480.

[4] Masaya Nakata and Will N Browne. 2020. Learning Optimality Theory for Accuracy-based Learning Classifier Systems. *IEEE Trans. Evol. Comput.* (2020).

[5] Mahmoud Noor. 2023. High-Resolution CatDogBird Image Dataset. https://www.kaggle.com/datasets/mahmoudnoor/high-resolution-catdogbird-image-dataset-13000. [Online; accessed Nov 12, 2023].

[6] Harisu Abdullahi Shehu, Abubakar Siddique, Will N Browne, and Hedwig Eisenbarth. 2021. Lateralized approach for robustness against attacks in emotion categorization from images. In *Int. Conf. on the App. of Evol. Comput. (Part of EvoStar)*. Springer, 469–485.

[7] Abubakar Siddique and Will Browne. 2022. Learning classifier systems: cognitive inspired machine learning for eXplainable AI. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*. 1081–1110.

[8] Abubakar Siddique, Will Browne, and Ryan Urbanowicz. 2023. Modern applications of evolutionary rule-based machine learning. In *Proceedings of the Companion Conference on Genetic and Evolutionary Computation*. 1301–1330.

[9] Abubakar Siddique, Will N Browne, and Gina M Grimshaw. 2020. Lateralized learning for robustness against adversarial attacks in a visual classification system. In *Proc. Gen. and Evol. Comput. Conf.* 395–403.

[10] Abubakar Siddique, Will N Browne, and Gina M Grimshaw. 2021. Frames-of-reference-based learning: Overcoming perceptual aliasing in multistep decision-making tasks. *IEEE Transactions on Evolutionary Computation* 26, 1 (2021), 174–187.

[11] Abubakar Siddique, Will N. Browne, and Gina M. Grimshaw. 2023. Lateralized Learning to Solve Complex Boolean Problems. *IEEE Transactions on Cybernetics* 53, 11 (2023), 6761–6775. doi:10.1109/TCYB.2022.3166119

[12] Abubakar Siddique, Muhammad Iqbal, and Will N Browne. 2016. A comprehensive strategy for mammogram image classification using learning classifier systems. In *Evol. Comput., 2016 IEEE Cong. on*. IEEE, 2201–2208.

[13] Abubakar Siddique, Muhammad Iqbal, and Will N Browne. 2024. Enhancing Visual Focus: Incorporating Attention into Rule-Based Machine Learning. In *2024 39th International Conference on Image and Vision Computing New Zealand (IVCNZ)*. IEEE, 1–6.

[14] Ryan J Urbanowicz and Jason H Moore. 2009. Learning classifier systems: a complete introduction, review, and roadmap. *Journal of Art. Evol. and Appl.* 2009 (2009), 1.

[15] Ryan J Urbanowicz and Jason H Moore. 2015. ExSTraCS 2.0: description and evaluation of a scalable learning classifier system. *Evolutionary intelligence* 8, 2-3 (2015), 89–116.