Neuro-Symbolic Learning from Temporal Sequences in Safety-Critical Systems

Luca Salvatore Lorello[a,b,*], Laura Carnevali[c], Marco Lippi[c] and Stefano Melacci[d]

[a]University of Modena and Reggio Emilia, Italy
[b]University of Pisa, Italy
[c]University of Florence, Italy
[d]University of Siena, Italy

Abstract. Safety-critical systems have seldom been used as an application domain for neuro-symbolic AI, despite their inherent characteristics that combine generation and processing of raw data, coming from heterogeneous devices, with enforcement or discovery of properties, usually encoded as rules or constraints. In this paper, we consider the task of classifying sequences of perceptual stimuli collected from a safety-critical system, where safety-related properties are represented in the form of linear temporal logic formulae. Our preliminary results on a benchmarking framework for temporal reasoning show that this kind of problem can be extremely challenging, for both neural-only and temporal neuro-symbolic approaches.

1 Introduction

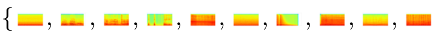
Neuro-Symbolic (NeSy) Artificial Intelligence (AI) aims to combine neural networks with symbolic approaches, with the goal of complementing the capability of the former to handle and learn from large data collections, with the expressivity of the latter in representing domain knowledge, typically in the form of rules, constraints, or logic facts [9]. Despite the growing interest in this research area, the successful applications of NeSy AI to real-world problems is widely recognized as one of the most urgent open challenges in the field [6]. Most of the existing literature in NeSy AI focuses on proof-of-concept implementations, proposed throughout the years in several domains, ranging from computer vision to knowledge base completion. Yet, the community is constantly looking for new benchmarks and tasks with a wider and stronger impact on society [17].

In this paper, we propose to consider the domain of safety-critical systems as a suitable scenario to test and apply NeSy techniques. Safety-critical systems (SCSs) are domains in which failure might produce significant damage to the system itself or to the environment, or even loss of life [13]. Typical application areas include healthcare, for critical tasks such as the monitoring of biomedical devices [24], transportation, as in the case of aircraft flight control systems [22], or space missions, for the detection of anomalies and cybersecurity vulnerabilities [25]. In most of these scenarios, safety must be ensured by assessing that the behaviour of the system is compliant with strict constraints, such as time constraints (e.g., tasks that must complete their executions within certain time limits) or logic constraints (e.g., events that must occur in a predefined sequential order) [11, 8]. Many formalisms can be used to model these systems, such as stochastic time Petri nets [3], deterministic or symbolic finite automata [10], linear temporal logic (LTL) [15], fault trees [20, 21] and others, with the aim of quantitatively evaluating dependability attributes [18, 4].

From the perspective of NeSy AI, SCSs represent an ideal setting to design novel benchmarks on real-world problems because it combines the availability of large data collections generated by physical devices interacting with the environment, with domain knowledge described via rules and constraints. Several NeSy tasks can be conceived in this setting, depending on whether domain knowledge is explicitly available or should rather be learned from examples, and the variables of interest are fully or partially observable [2]. In this paper, we propose to exploit NeSy AI approaches in the context of SCSs, in particular for the task of classifying sequences of perceptual stimuli according to their compliance to a certain LTL formula [14].

2 Case study

Consider an SCS made of two devices (A and B) characterized by ten possible states $\{Y_0, \ldots, Y_9\} \in \mathcal{Y}$, each associated with a perceptual signature, in the form of audio spectrograms $\{X_0, \ldots, X_9\} \in \mathcal{X}$. Suppose this system must comply to a *liveness* property [1] $\mathcal{F}$, asserting that an event p ("the sensor is in state 4") registered by sensor A must always be followed by another event q ("the sensor is in state 7"), observed by sensor B. Events can be tracked, by systematically evaluating the validity of a set of relational predicates $\mathcal{C}$, on the state of the system over time. The behavior described above can be represented compactly by the following specification:

$\mathcal{X}: = \{$  $\}$

$$\mathcal{Y}: = \{Y_0, \ldots, Y_9\}$$
$$\mathcal{C}: = \{\mathrm{p}(Z) : Z = Y_4,\ \mathrm{q}(Z) : Z = Y_7\}$$
$$\mathcal{F}: = \Box(\mathrm{p}(A) \rightarrow \Diamond \mathrm{q}(B))$$

This specification corresponds to an LTLZinc [14] problem, where $\mathcal{C}$ is encoded as MiniZinc [19] constraints and $\mathcal{F}$ is a linear temporal logic formula over finite domains (LTL_f) [5]. This framework enables the design of experiments in different learning and reasoning scenarios, depending on what kind of knowledge is available at training time, and which element of the tuple $\langle \mathcal{X}, \mathcal{Y}, \mathcal{C}, \mathcal{F} \rangle$ constitutes the learning objective. For example, in many cases, an SCS is designed with predefined specifications in mind, and it is therefore reasonable to assume prior knowledge about both $\mathcal{C}$ and $\mathcal{F}$, but not the mapping

* Corresponding Author. Email: luca.lorello@phd.unipi.it

Pattern	Category	Best Epoch	Avg Accuracy ↑	IC Accuracy ↑	CC Accuracy ↑	NSP Accuracy ↑	SC Accuracy ↑
IMMEDIATE FAILURE	NeSy	5	**0.73** ± 0.04	0.78 ± 0.04	**0.87** ± 0.02	**0.63** ± 0.05	**0.63** ± 0.05
	Neural	5	0.68 ± 0.00	**0.80** ± 0.01	0.86 ± 0.01	0.57 ± 0.00	0.50 ± 0.00
	Random		0.36	0.10	0.50	0.33	0.50
LIVENESS	NeSy	2	**0.76** ± 0.01	**0.80** ± 0.01	**0.83** ± 0.01	**0.69** ± 0.00	**0.71** ± 0.01
	Neural	5	0.68 ± 0.01	0.76 ± 0.02	0.72 ± 0.01	0.64 ± 0.01	0.59 ± 0.02
	Random		0.40	0.10	0.50	0.50	0.50
REAL-TIME RESPONSE	NeSy	3	**0.66** ± 0.02	**0.80** ± 0.00	**0.85** ± 0.01	**0.48** ± 0.00	**0.52** ± 0.07
	Neural	6	0.59 ± 0.02	0.76 ± 0.06	0.76 ± 0.02	0.35 ± 0.02	0.50 ± 0.00
	Random		0.34	0.10	0.50	0.25	0.50

Table 1. Best results ($mean \pm std$ over 3 replicates) on the proposed tasks. Random indicates a baseline with random predictions at each stage.

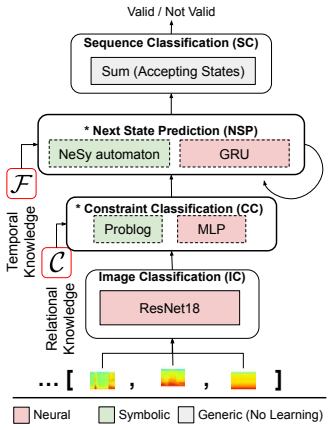


Figure 1. Multi-stage architecture for temporal reasoning. Each block can be instantiated in either neural or symbolic flavors. Adapted from [14].

$\mathcal{X} \mapsto \mathcal{Y}$. Within this setting, a *sequence classification* task corresponds to predict whether a given sequence of perceptual stimuli $\mathcal{X}^t$ along a discrete set of n timesteps $t = \{1, \ldots, n\}$ satisfies $\mathcal{F}$ or not, and it corresponds to the *verification* of $\mathcal{F}$ within a given sequence $\mathcal{X}^t$, by neuro-symbolic means. In other cases, either $\mathcal{C}$ or $\mathcal{F}$ could be unknown, and the sequence classification task would thus involve the *induction* of temporal safety properties directly from system traces: in these settings, the NeSy system is trained to discriminate between positive and negative sequences, without knowing neither $\mathcal{F}$ nor $\mathcal{C}$.

3 Methodology

We address the sequence classification task exemplified in Section 2 by cascading multiple decisions, following the approach described in [14]. In particular, we employ a multi-stage pipeline (Figure 1) composed of the following sub-tasks: (IC) image classification, mapping data from each $\mathcal{X}_i$ to the corresponding $\mathcal{Y}_i$; (CC) constraint classification, leveraging relational knowledge $\mathcal{C}$; (NSP) next state prediction, leveraging temporal knowledge $\mathcal{F}$; (SC) sequence classification, i.e., the final decision. Each stage i is associated with a loss function, weighted by a corresponding hyper-parameter λ_i. IC and NSP are trained by means of categorical cross-entropy, while CC and SC employ a binary cross-entropy loss.

4 Experiments

Using the LTLZinc framework,[1] we generate three tasks, following well-known LTL patterns for safety-critical applications [7]. For each task, we assume three spectrogram images X, Y, Z, and the following constraint mapping $\mathcal{C}$:

$$\begin{aligned}\mathcal{C}\colon\quad & \mathtt{p}(X,Y,Z) : (X+Y) \equiv Z \mod 10;\\ & \mathtt{q}(X,Y,Z) : \mathtt{all\_different}([X,Y,Z]);\\ & \mathtt{r}(X,Y,Z) : (X<Y<Z) \vee (X>Y>Z);\\ & \mathtt{s}(X,Y,Z) : X \neq Z \wedge (X=Y \vee Y=Z).\end{aligned}$$

Each task corresponds to a different safety-critical property $\mathcal{F}$:

IMMEDIATE FAILURE p is false after r:
$\square(\mathtt{r} \rightarrow \square\neg\mathtt{p})$;

LIVENESS s always follows p:
$\square(\mathtt{p} \rightarrow \Diamond\mathtt{s})$;

REAL-TIME RESPONSE s responds to p between q and r:
$\square((\mathtt{q} \wedge \bigcirc\Diamond\mathtt{r}) \rightarrow (\mathtt{p} \rightarrow (\neg\mathtt{r}\,\mathcal{U}\,(\mathtt{s} \wedge \neg\mathtt{r})))\,\mathcal{U}\,\mathtt{r})$.

Datasets contain 1000 sequences (800 train, 100 validation, 100 test samples) of random length between 10 and 25 timesteps. Each timestep is associated with three RGB images sampled from the UrbanSound-Spectrogram dataset,[2] resized to fit into an 224×224 image with white background. Images are augmented during training and inference, according to the original ResNet18 transforms [12]. The modular architecture of Figure 1 is initialized in two flavors: NEURAL (ResNet18, Multi-layer Perceptron, Gated Recurrent Unit, red dashed blocks), and NESY (ResNet18, Deep Problog [16], NeSy Automaton [23], green dashed blocks). After selecting optimal hyper-parameters (optimizer: Adam, learning rate: 10^{-4}, MLP: 64 neurons, GRU: 64 hidden units) on the simplified task (the one introduced in Sec. 2), we initialize the IC module with one epoch of pre-training on image labels only, then supervise every stage ($\lambda_{\text{CC}} = \lambda_{\text{NSP}} = \lambda_{\text{SC}} = 1.0, \lambda_{\text{IC}} = 0.1$) for a maximum of 7 epochs.

Table 1 summarizes the results on the test set, evaluated on the best-performing epoch (selected by average accuracy across all modules, measured on the validation set). In general, it can be observed that, although both NEURAL and NESY methods reach similar image classification performance, downstream objectives become increasingly difficult for the Neural-only method. Overall, the NESY method achieves the best performance across all symbolic objectives for every task, even though there is margin for improvement. Knowledge availability can only partially compensate the challenging nature of this setting, as even NESY approaches struggle with harder formulae. Training behavior (not shown) indicates that this effect is caused by severe over-fitting at the NSP stage, in spite of good upstream generalization of the CC objective, highlighting optimization challenges of the NeSy Automaton module, in spite of full supervisions available.

[1] https://github.com/continual-nesy/LTLZinc

[2] https://github.com/mashrin/UrbanSound-Spectrogram

These preliminary experiments show that, while significantly outperforming knowledge-agnostic, neural-only methods, in every explored task, and in spite of perfect knowledge availability, NeSy performance quickly degrades as temporal behavior increases in complexity. To successfully deploy NeSy temporal reasoners in safety-critical settings, it is crucial to boost their performance in a way which is not affected by temporal complexity, especially for real-world settings, where full supervisions are not available.

Acknowledgements

L.S.L. scholarship was funded by the Italian Ministry of University and Research (DM 351/2022, PNRR). L.C. was supported by the European Union under the Italian National Recovery and Resilience Plan (NRRP) of NextGenerationEU, partnership on "Telecommunications of the Future" (PE00000001 - program "RESTART"). M.L. was supported by CAI4DSA actions (Collaborative Explainable neuro-symbolic AI for Decision Support Assistant), PARTENARIATO ESTESO "Future Artificial Intelligence Research - FAIR", SPOKE 1 "Human-Centered AI" Università di Pisa, CUP B13C23005640006. The scholarship by L.S.L. was funded by the Italian Ministry of University and Research (DM 351/2022, PNRR). S.M. was supported by the University of Siena (Piano per lo Sviluppo della Ricerca - PSR 2024, F-NEW FRONTIERS 2024), under the project "TIme-driveN StatEful Lifelong Learning" (TINSELL) and also by the project "CONSTR: a COllectionless-based Neuro-Symbolic Theory for learning and Reasoning", PARTENARIATO ESTESO "Future Artificial Intelligence Research - FAIR", SPOKE 1 "Human-Centered AI" Università di Pisa, "NextGenerationEU", CUP I53C22001380006.

References

[1] B. Alpern and F. B. Schneider. Defining liveness. *Information processing letters*, 21(4):181–185, 1985.

[2] L. Carnevali and M. Lippi. Neuro-symbolic artificial intelligence for safety engineering. In *International Conference on Computer Safety, Reliability, and Security*, pages 438–445. Springer, 2024.

[3] L. Carnevali, L. Ridi, and E. Vicario. A quantitative approach to input generation in real-time testing of stochastic systems. *IEEE Transactions on Software Engineering*, 39(3):292–304, 2013.

[4] L. Carnevali, S. Cerboni, L. Montecchi, and E. Vicario. FaultFlow: an MDE Library for Dependability Evaluation of Component-Based Systems. *IEEE Transactions on Dependable and Secure Computing*, 22(4): 3431–3448, 2025.

[5] G. De Giacomo and M. Y. Vardi. Linear temporal logic and linear dynamic logic on finite traces. In *Ijcai*, volume 13, pages 854–860, 2013.

[6] L. De Raedt, S. Dumančić, R. Manhaeve, and G. Marra. From statistical relational to neural-symbolic artificial intelligence. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, pages 4943–4950, 2021.

[7] M. B. Dwyer, G. S. Avrunin, and J. C. Corbett. Property specification patterns for finite-state verification. In *Proceedings of the second workshop on Formal methods in software practice*, pages 7–15, 1998.

[8] F. M. Favarò and J. H. Saleh. Application of temporal logic for safety supervisory control and model-based hazard monitoring. *Reliability Engineering & System Safety*, 169:166–178, 2018.

[9] A. d. Garcez and L. C. Lamb. Neurosymbolic ai: The 3 rd wave. *Artificial Intelligence Review*, 56(11):12387–12406, 2023.

[10] G. Giantamidis, S. Basagiannis, and S. Tripakis. Efficient translation of safety ltl to dfa using symbolic automata learning and inductive inference. In *International Conference on Computer Safety, Reliability, and Security*, pages 115–129. Springer, 2020.

[11] P. Graydon and I. Bate. Realistic safety cases for the timing of systems. *The Computer Journal*, 57(5):759–774, 2014.

[12] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[13] J. C. Knight. Safety critical systems: challenges and directions. In *Proceedings of the 24th international conference on software engineering*, pages 547–550, 2002.

[14] L. S. Lorello, M. Lippi, and S. Melacci. A neuro-symbolic framework for sequence classification with relational and temporal knowledge. *International Joint Conference on Artificial Intelligence (IJCAI)*, 2025.

[15] W.-g. Ma and X.-h. Hei. An approach for design and formal verification of safety-critical software. In *2010 International Conference on Computer Application and System Modeling (ICCASM 2010)*, volume 4, pages V4–264. IEEE, 2010.

[16] R. Manhaeve, S. Dumancic, A. Kimmig, T. Demeester, and L. De Raedt. Deepproblog: Neural probabilistic logic programming. *Advances in neural information processing systems*, 31, 2018.

[17] R. Manhaeve, F. Giannini, M. Ali, D. Azzolini, A. Bizzarri, A. Borghesi, S. Bortolotti, L. De Raedt, D. Dhami, M. Diligenti, et al. Benchmarking in neuro-symbolic ai. In *Proceedings of The 4th International Joint Conference on Learning & Reasoning*, 2024.

[18] A. Maurya and D. Kumar. Reliability of safety-critical systems: A state-of-the-art review. *Quality and Reliability Engineering International*, 36 (7):2547–2568, 2020.

[19] N. Nethercote, P. J. Stuckey, R. Becket, S. Brand, G. J. Duck, and G. Tack. Minizinc: Towards a standard cp modelling language. In *International Conference on Principles and Practice of Constraint Programming*, pages 529–543. Springer, 2007.

[20] M. Roth and P. Liggesmeyer. Modeling and analysis of safety-critical cyber physical systems using state/event fault trees. In *SAFECOMP 2013-Workshop DECS (ERCIM/EWICS Workshop on Dependable Embedded and Cyber-physical Systems) of the 32nd International Conference on Computer Safety, Reliability and Security*, page NA, 2013.

[21] E. Ruijters and M. Stoelinga. Fault tree analysis: A survey of the state-of-the-art in modeling, analysis and tools. *Computer science review*, 15: 29–62, 2015.

[22] A. J. Stolzer, R. L. Sumwalt, and J. J. Goglia. *Safety management systems in aviation*. CRC Press, 2023.

[23] E. Umili, R. Capobianco, and G. De Giacomo. Grounding ltlf specifications in image sequences. In *Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning*, volume 19, pages 668–678, 2023.

[24] V. Vakhter, B. Soysal, P. Schaumont, and U. Guler. Threat modeling and risk analysis for miniaturized wireless biomedical devices. *IEEE Internet of Things Journal*, 9(15):13338–13352, 2022.

[25] L. Vessels, K. Heffner, and D. Johnson. Cybersecurity risk assessment for space systems. In *2019 IEEE Space Computing Conference (SCC)*, pages 11–19. IEEE, 2019.