|  |  |
| --- | --- |
| cetlogo ***CHEMICAL ENGINEERING TRANSACTIONS***  ***VOL. xxx, 2025*** | A publication of  aidiclogo_grande |
| The Italian Association  of Chemical Engineering  Online at www.cetjournal.it |
| Guest Editors: Bruno Fabiano, Valerio Cozzani  Copyright © 2025, AIDIC Servizi S.r.l. **ISBN** 979-12-81206-xx-y; **ISSN** 2283-9216 | |

On Neuro-Symbolic AI for Abnormal Event Detection in Process Safety

Julien Amblarda,\*, Alessandra Russob, Gürkan Sinc

aImperial College, South Kensington, London, United Kingdom

bImperial College, South Kensington, London, United Kingdom

cTechnical University of Denmark, Kongens Lyngby, Denmark

julien.amblard16@imperial.ac.uk

In recent years, Artificial Intelligence (AI) techniques have led to numerous applications, from self-driving cars, to chatbots and crop monitoring. However, due to the high-risk environments in which they work, many safety-critical fields have fallen back on adopting these new methods. To be considered for use in such hazardous settings, an AI system would need to be capable of reasoning while fully taking into account relevant knowledge of the domain it is applied to, as well as being able to provide a clear explanation for each conclusion it makes. With this in mind, Neuro-Symbolic Learning — a hybrid AI approach combining Neural Networks with logic-based reasoning — shows much potential for use in Process Safety.

This paper aims to provide the reader with a survey of the state-of-the-art Neuro-Symbolic approaches, with an emphasis on abnormal events. Additional topics of interest will be AI safety and Explainable AI. The paper concludes by setting the scene for future research focused specifically on abnormal event detection in chemical processes, suggesting novel frameworks that could be developed and used in real-time applications. By the end of this paper, the reader should have a rough idea of what Neuro-Symbolic Learning entails, and how it can be used to model complex systems in the context of Process Safety.

* 1. Introduction

Neuro-Symbolic Learning is a hybrid Artificial Intelligence (AI) method that combines Neural Networks (NNs) with logic-based reasoning into a single framework, to take advantage of the benefits of both approaches. While NNs can work with vast amounts of potentially noisy data, automatically picking up on minute nuances, they require substantial computational resources, are prone to hallucinations, and are effectively “black box” systems. In contrast, Symbolic Learning systems are based on formal logic and can be instructed to rigorously follow a set of constraints, while drawing optimal conclusions from far fewer data than NNs. However, they do not scale easily and cannot cope with data whose format is not as expected. In the paper we provide a high-level overview of the Neuro-Symbolic Learning research area, starting from the first principles of logic.

* 1. Formal logic and logic-based learning

To properly understand Symbolic Learning, one must first get a basic grasp of Formal Logic, Classical Logic being the most widely used type. Within Classical Logic the most fundamental subtype is Propositional Logic (Lifschitz et al., 2008). Its syntax is made of symbols called Atoms — whose individual Truth Value is either True or False, as indicated by the Interpretation — and logical Connectives (e.g., “or”, “and”, “implies”). Table 1 shows the semantics of the main Connectives used in Propositional Logic.

Table 1: The semantics of logical Connectives in Propositional Logic, for Atoms X and Y

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| X | Y | not X | X and Y | X or Y | X implies Y | X if and only if Y |
| True | True | False | True | True | True | True |
| False | True | True | False | True | True | False |
| True | False | False | False | True | False | False |
| False | False | True | False | False | True | True |

First-Order Logic (FOL) builds on Propositional Logic by diversifying the categories of symbols: when given their required number of arguments, Predicate Symbols are evaluated to Truth Values, whereas Function Symbols map to either Constants or Variables (Lifschitz et al., 2008). FOL also adds two new operators: the Existential Quantifier (“for some assignment of Variable x, Formula F holds”) and the Universal Quantifier (“for all assignments of x, F holds”). Using FOL, one can model a problem by specifying the Facts and Rules that define the Background Knowledge (KB) of the corresponding Learning Task. Facts are essentially parametrized Predicates that convey irrefutable knowledge, and Rules lay out the conclusions that can be made conditioned on given prior information. Table 2 shows a set of Facts and Rules along with their meaning in English. By combining KB with a set of observations, one can make inferences. In Formal Logic, there are three main ways to reason about a Learning Task (Minnameier, 2010). Deduction is simply based on iteratively feeding the Facts and observations of a problem into its Rules (e.g., “if all chemists wear gloves and all gloves are blue, then Charlie who is a chemist wears blue gloves”). Inductive reasoning is based on the idea of generalizing a set of observations while remaining consistent with any pre-existing knowledge (“if all observed chemists are known to work only during the day, then it can be hypothesized that *all* chemists work only during the day”). Finally, abduction involves making conjectures about missing Facts that could explain a set of observations given some general rules (“if a kettle is plugged in and steam starts to come out of its spout, it can be hypothesized that the kettle is on”, given the general rule “if a kettled is plugged in and it is on then steam will come out of its spout”).

Table 2: Examples of Formulae in FOL

|  |  |
| --- | --- |
| Formula | Meaning in English |
|  | A cooling water feed is an input to the reactor |
|  | “sensor26” is a thermometer |
|  | “sensor26” connects to the cooling water feed |
|  | Every input to the reactor has a thermometer |
|  | Reducing the flow rate of the cooling water feed causes the reactor to warm up |

Aside from Classical Logic there are various classes of formal logic, one of which is called Temporal Logic (Venema, 2017). As an extension of Propositional Logic, its Interpretations are supplemented with a discrete temporal dimension allowing expressions of the form “p will be true at some point in the future, and once this happens q will be true forever”. To extend this further, Signal Temporal Logic (STL) uses a continuous measure of time and provides syntax to limit the temporal operators to specific time ranges (Maler and Nickovic, 2004).

Table 3: An ASP program defining how to modulate the temperature around a reactor based on reaction type

|  |  |
| --- | --- |
| Program | Answer sets |
|  |  |

Classical Logic is Monotonic i.e., new information cannot lead to refuting a claim which was previously sound (McCarthy, 1980). Humans do not reason in this manner, which motivates a class of logics that are Non-Monotonic. In such logics, one can use Negation As Failure (NAF) instead of strong negation, which succeeds when there is no proof that leads to its argument being true (Clark, 1977). In particular, Answer Set Programming (ASP) is a technique representing domain knowledge and Facts to perform inference in order to solve complex reasoning tasks (Lifschitz, 2019). It is important to note that “Answer Sets” (i.e., solutions to given problems) are not always unique as ASP can provide syntax for representing Choice Rules (i.e., non-deterministic behaviors), letting the solver pick between one or more expressions to set to true which leads to alternative inferences. Table 3 shows an ASP program along with its answer sets.

ASP has been extended to reason inductively, thanks to symbolic learning frameworks like ILASP (Law et al., 2018) and FastLAS (Law et al., 2021). Moreover, recent versions can learn in the presence of noise, by making judgments coherent with as many of the provided Examples (i.e., contextualized observations) as possible.

* 1. Neuro-symbolic AI
     1. General overview and survey

Neuro-Symbolic AI is a very diverse field due to the many ways the two AI approaches can be combined. Wan et al. (2024a) identified five different paradigms: the first one is defined by a symbolic reasoner that invokes one or more NNs as part of its inference process; the second is characterized by a pipeline where the symbolic and neural processes are not interleaved; and the other three paradigms use logic formulae or calls to a symbolic reasoner as an integral part of NNs. Ideally the combination of neural and symbolic approaches is symbiotic; the neural component helps with learning from messy and/or noisy data, while the symbolic component encodes domain knowledge to constrain the learning process of the neural component. Note that modern NNs take advantage of GPU acceleration, while symbolic reasoners limit themselves to CPU and memory utilization (Wan et al., 2024b). This difference in computational needs must be taken into account especially when scaling a framework, though sometimes parallelization can help alleviate performance bottlenecks.

To start off with a Neuro-Symbolic framework that is pipeline-based, DeepProbLog (Manhaeve et al., 2018) extends a probabilistic logic programming language called ProbLog whose syntax is roughly the same as ASP’s, but allows specifying probabilistic facts. More specifically, facts in DeepProbLog can be represented as an exclusive disjunction of weighted choices, whose distribution is estimated by training a neural classifier. When performing deductive inference, the probability of consequences is computed from complex formulae by recursing upwards from the smallest sub-formulae and probabilistic facts. Figure 1 shows a neuro-symbolic learning task, based on DeepProbLog, where the head of the rule in line 2 is determined by a neural prediction. NeurASP (Yang et al., 2023) is similar but builds on top of ASP. In both cases training the neural component goes hand in hand with updating the probabilities of a symbolic program, leaving at the end trained networks for predicting features from data, and these can be used elsewhere. What’s more, the two frameworks needed almost exactly the same number of iterations to converge to the same accuracy on a set of test problems. In both cases, however, a risk is that suboptimal NN performance can lead to probabilistic programs being altered to the point where the most likely conclusions no longer make sense in the context of the problem.

|  |  |
| --- | --- |
| 1 |  |
| 2 |  |
| 3 |  |
| 4 |  |
| 5 |  |
| 6 |  |
| 7 |  |

*Figure 1: A DeepProbLog program by* Manhaeve et al. (2018) to reason about the likelihood of winning a game, while training a neural model to find the optimal distribution for the bias of a coin

Moving on to a Neuro-Symbolic framework whose main architecture is neural, Dong et al. (2019) propose Neural Logic Machines (NLMs) which are NNs that replicate the process of deductive reasoning in FOL. Every layer in a NLM intuitively represents the application of a single rule. At each neuron, there is a Multi-Layer Perceptron (MLP) that takes in the probabilities of all instantiations for every condition of the layer’s rule. The result of applying the MLP is then fed to the sigmoid function that turns it into a value between 0 and 1, which can be rounded to obtain a Truth Value. The authors have experimented with problems ranging from learning relationships between family members to sorting algorithms, where NLMs achieved perfect accuracy. Dong et al. show that their framework can easily scale after being trained on a small version of a problem, though the number of predicate instantiations should stay low to avoid the dimensions of tensors in the network from getting out of hand. Riegel et al. (2020) approach this task differently by defining a Logical Neural Network (LNN) as an amalgam of syntax trees of probabilistic formulae, where leaf nodes are predicates and non-leaf nodes are logical operators. Unlike NLMs, inference can occur top-down or bottom-up, making it a more powerful framework for theorem proving, and allowing for both deductive and inductive reasoning. Moreover, the probability of a sub-formula is given by a subset of the range [0, 1] rather than a single probability value, providing LNNs with the ability to express uncertainty. In both frameworks the result is a NN that constructs the most likely formulae for a given situation, instead of filling in the gaps for a preexisting logic program as was the case above.

* + 1. Anomaly detection in complex systems

In recent years, research efforts have led to Neuro-Symbolic frameworks for anomaly detection, the most relevant of which will be discussed in what follows. Firstly, Capogrosso et al. (2023) focus on embedded systems and approach the problem as an “Out-Of-Distribution (OOD) classification” task, that is, an anomaly is detected when their Diffusion Model (Ho et al., 2020) trained on “normal” signals cannot reconstruct a given input signal. They combine this with a set of constraints in FOL, incurring a penalty in the Diffusion Model’s loss function when these are not satisfied. They then condense the model using clever tricks to run their framework in real-time on embedded systems. Aravanis and Kabouris (2022) look at the voltage waveforms of power grids to match specific patterns to particular anomaly classes. In essence, they use a Convolutional Neural Network to classify a waveform, followed by the use of handcrafted ASP rules to disambiguate the NN’s output when it does not achieve high enough confidence to pick a unique anomaly class. Bohne et al. (2023) use Knowledge Graphs to encode domain knowledge about On-Board Diagnostics in the automotive domain. This has the benefit of linking related concepts spatially and providing a built-in visualization tool. (See Figure 2 for an example of a knowledge graph.) They combine this with a Fully Convolutional Network that classifies sensor oscillograms, using Class Activation Maps to highlight where in an oscillogram an anomaly was identified. If the knowledge graph contains enough information to help debug this anomaly, it will produce a root cause hypothesis and share its reasoning with a mechanic; otherwise, humans will diagnose the fault and add the associated knowledge to the graph for future diagnoses. Finally, Tian et al. (2024) have used an extension of STL to represent deviations from the norm in the vibration signals of bearings. More specifically, their solution involves training a NN to recognize these deviations and associate each type of fault with a formula describing the order in which the deviations need to occur. They have also devised a graph-based representation for their formulae, which shows the chronological relationships between the points where deviations are expected to occur.

A group of blue circles and white circles

Description automatically generated with medium confidence

*Figure 2: A Knowledge Graph for* On-Board Diagnostics (Bohne et al., 2023)

Despite the many differences between these four state-of-the-art frameworks, they all use a NN to produce simple fragments of information from complex and sometimes messy data. For all but the last framework, this information is then put together with domain-specific knowledge for a symbolic reasoner, entrusted with making a final judgment. Overall, it can be observed that explainability increases when symbolic knowledge is used to constrain what a NN learns during training, and interpretability increases when the symbolic part has a more direct influence on drawing conclusions (or the framework outputs formulae). Providing a visual representation of the system’s reasoning process or outputs also respectively helps with explainability and interpretability.

* 1. Safe and trusted AI for industrial processes

When using an AI-based framework to make important decisions, it is crucial to understand how it reasons. As hinted at earlier, being able to both explain how a judgment was reached and make this judgment easy to interpret by humans goes a long way in establishing and maintaining trust. More to the point, good practices in explainable AI (XAI) go together with effective “human-agent interaction” (Miller, 2019), where the word “Agent” denotes an intelligent framework that can provide explanations to humans. In Miller’s findings, replicating the communication patterns that are common between humans could be key to unlocking high standards of XAI. Rather than relying on humans to decipher an Agent’s explanation, the two parties should come to a mutual understanding through discussion. Depending on the context in which a question is asked, an Agent should be able to interpret it accordingly and provide a straightforward explanation with meaningful new information.

In complex industrial processes (e.g., chemical processes), where operators often work in high-risk environments, good XAI practices must be followed if an Agent is to be trusted and therefore used as intended. Moreover, real-time feedback from an Agent should be available to operators who need help in making a decision (“AI-in-the-loop”), and operators should be able to supervise the parts of a process controlled by Agents (“human-in-the-loop”) (Arunthavanathan et al., 2024).

* 1. Future work

To fulfill the objective of running chemical processes more safely and with less manual intervention, further research should be carried out in the field of Neuro-Symbolic AI for abnormal event detection. The first step should be to collect a variety of domain-specific knowledge, from information about the relevant chemical reactions, to the layout of components and causal relationships in processes. This KB should be combined with examples of normal and abnormal processes modeled in logic. Then, inductive inference can be used to learn rules that allow identifying and even predicting abnormal process behaviors. An important part of this system will be NN-based; this will help the symbolic inference module find the relevant patterns in the anomalous process data, and likely be useful for incorporating new predictive knowledge into the KB. Once a sufficiently cohesive framework has been developed, it can be deployed in real-world settings such as hydrogen energy production systems. An illustration of such an approach is presented in Figure 3.

However, this research does not come without its challenges. In practice, process plants are relatively complex and operate with nonlinear dynamics i.e., abnormal events could have multiple causal pathways. To make matters worse, historical process data are not readily found and may be incomplete or erroneous. Faced with these complexities, attaining a maximally accurate neuro-symbolic model will require building it up step-by step, starting from single-component modeling and data acquisition from a well-defined simulation.



Learn Rules

Scale framework

Apply to a real-world problem: green hydrogen energy production?

NN:

Deep Learning?

Support Vector Machine?

Test

Induction-based symbolic reasoning:

ILASP?

FastLAS?

Process KB

Normal/abnormal event examples

program.las

—

1

2

3

NN?

Write model

Gather data

*Figure 3: A diagram showing the main steps of the proposed work plan*

* 1. Conclusion

In this paper, symbolic inference is introduced, leading to a discussion on Neuro-Symbolic AI — a promising research area both in terms of performance and robustness. A summary of different types of data representation, types of symbolic reasoning, NN architectures, and Neuro-Symbolic paradigms was given — these come with varying levels of explainability and interpretability. An overview of recent neuro-symbolic AI approaches for anomaly detection is discussed, with an emphasis on industrial processes whose high-risk environments call for paying specific attention to the communication abilities of Agents in the context of XAI. Finally, this paper outlines a proposal for future work centered on Neuro-Symbolic AI for anomaly detection in chemical processes.

References

Aravanis, T. and Kabouris, I., 2022, November. A Neuro-Symbolic Approach for Fault Diagnosis in Smart Power Grids. In Proceedings of the 26th Pan-Hellenic Conference on Informatics (pp. 90-95).

Arunthavanathan, R., Sajid, Z., Khan, F. and Pistikopoulos, E., 2024. Artificial intelligence–Human intelligence conflict and its impact on process system safety. Digital Chemical Engineering, 11, p.100151.

Badreddine, S., Garcez, A.D.A., Serafini, L. and Spranger, M., 2022. Logic tensor networks. Artificial Intelligence, 303, p.103649.

Bohne, T., Windler, A.K.P. and Atzmueller, M., 2023, December. A Neuro-Symbolic Approach for Anomaly Detection and Complex Fault Diagnosis Exemplified in the Automotive Domain. In Proceedings of the 12th Knowledge Capture Conference 2023 (pp. 35-43).

Capogrosso, L., Mascolini, A., Girella, F., Skenderi, G., Gaiardelli, S., Dall'Ora, N., Ponzio, F., Fraccaroli, E., Di Cataldo, S., Vinco, S. and Macii, E., 2023, September. Neuro-Symbolic Empowered Denoising Diffusion Probabilistic Models for Real-Time Anomaly Detection in Industry 4.0: Wild-and-Crazy-Idea Paper. In 2023 Forum on Specification & Design Languages (FDL) (pp. 1-4). IEEE.

Clark, K.L., 1977. Negation as failure. In Logic and data bases(pp. 293-322). Boston, MA: Springer US.

Dong, H., Mao, J., Lin, T., Wang, C., Li, L. and Zhou, D., 2019. Neural logic machines. arXiv preprint arXiv:1904.11694.

Ho, J., Jain, A. and Abbeel, P., 2020. Denoising diffusion probabilistic models. Advances in neural information processing systems, 33, pp.6840-6851.

Law, M., Russo, A. and Broda, K., 2018. Inductive learning of answer set programs from noisy examples. arXiv preprint arXiv:1808.08441.

Law, M., Russo, A., Broda, K. and Bertino, E., 2021. Scalable Non-observational Predicate Learning in ASP. In IJCAI (pp. 1936-1943).

Lifschitz, V., 2008, July. What is answer set programming?. In Proceedings of the 23rd national conference on Artificial intelligence-Volume 3 (pp. 1594-1597).Lifschitz, V., Morgenstern, L. and Plaisted, D., 2008. Knowledge representation and classical logic. Foundations of Artificial Intelligence, 3, pp.3-88.

Maler, O. and Nickovic, D., 2004, September. Monitoring temporal properties of continuous signals. In International symposium on formal techniques in real-time and fault-tolerant systems (pp. 152-166). Berlin, Heidelberg: Springer Berlin Heidelberg.

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

McCarthy, J., 1980. Circumscription—a form of non-monotonic reasoning. Artificial intelligence, pp.27-39.

Miller, T., 2019. Explanation in artificial intelligence: Insights from the social sciences. Artificial intelligence, 267, pp.1-38.

Minnameier, G., 2010. The logicality of abduction, deduction, and induction. In Ideas in action: Proceedings of the applying Peirce conference (pp. 239-251). Helsinki: Nordic Pragmatism Network.

Riegel, R., Gray, A., Luus, F., Khan, N., Makondo, N., Akhalwaya, I.Y., Qian, H., Fagin, R., Barahona, F., Sharma, U. and Ikbal, S., 2020. Logical neural networks. arXiv preprint arXiv:2006.13155.

Tian, R., Cui, M. and Chen, G., 2024. A neural-symbolic network for interpretable fault diagnosis of rolling element bearings based on temporal logic. IEEE Transactions on Instrumentation and Measurement.

Venema, Y., 2017. Temporal logic. The Blackwell guide to philosophical logic, pp.203-223. Maldon, MA, US.

Wan, Z., Liu, C.K., Yang, H., Li, C., You, H., Fu, Y., Wan, C., Krishna, T., Lin, Y. and Raychowdhury, A., 2024. Towards cognitive ai systems: a survey and prospective on neuro-symbolic ai. arXiv preprint arXiv:2401.01040.

Wan, Z., Liu, C.K., Yang, H., Raj, R., Li, C., You, H., Fu, Y., Wan, C., Samajdar, A., Lin, Y.C. and Krishna, T., 2024, May. Towards cognitive ai systems: Workload and characterization of neuro-symbolic ai. In 2024 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)(pp. 268-279). IEEE.

Yang, Z., Ishay, A. and Lee, J., 2023. Neurasp: Embracing neural networks into answer set programming. arXiv preprint arXiv:2307.07700.