Neuro-Symbolic techniques for Predictive Maintenance

(Discussion Paper)

Angelica Liguori[1], Simone Mungari[1,*], Ettore Ritacco[2], Francesco Ricca[1], Giuseppe Manco[3] and Salvatore Iiritano[4,†]

[1]*University of Calabria, Via P. Bucci, Rende, 87036, Italy*
[2]*University of Udine, Via Palladio, Udine, 33100, Italy*
[3]*ICAR-CNR, Via P. Bucci 8-9/C, Rende, 87036, Italy*
[4]*Revelis S.r.l., Viale della Resistenza, Rende, 87036, Italy*

Abstract

Predictive maintenance plays a key role in the core business of the industry due to its potential in reducing unexpected machine downtime and related cost. To avoid such issues, it is crucial to devise artificial intelligence models that can effectively predict failures. Predictive maintenance current approaches have several limitations that can be overcome by exploiting hybrid approaches such as Neuro-Symbolic techniques. Neuro-symbolic models combine neural methods with symbolic ones leading to improvements in efficiency, robustness, and explainability. In this work, we propose to exploit hybrid approaches by investigating their advantage over classic predictive maintenance approaches.

Keywords

Predictive Maintenance, Neuro-Symbolic, Root Cause Analysis, Logic programming, Data-driven, Model-based

1. Introduction

Nowadays Artificial Intelligence (AI) has attracted a lot of attention in several application domains, including industry. In particular, predictive maintenance plays a key role, since it allows industries to preventively avoid internal systems failures, as well as reduce costs for service interruptions. Model-based and data-driven [1] strategies are currently exploited to develop design, optimization, diagnostic, and maintenance phases. Model-based techniques exploit mathematical models as well as background knowledge derived from human experts. Mathematical models allow describing relationships that govern a determined environment. Vice versa, data-driven approaches are inductive methods: models are created by generalizing

SEBD 2023: 31st Symposium on Advanced Database System, July 02–05, 2023, Galzignano Terme, Padua, Italy
*Corresponding author.
†These authors contributed equally.
angelica.liguori@dimes.unical.it (A. Liguori); simone.mungari@unical.it (S. Mungari); ettore.ritacco@uniud.it (E. Ritacco); francesco.ricca@unical.it (F. Ricca); giuseppe.manco@icar.cnr.it (G. Manco); salvatore.iiritano@revelis.eu (S. Iiritano)
0000-0001-9402-7375 (A. Liguori); 0000-0002-0961-4151 (S. Mungari); 0000-0003-3978-9291 (E. Ritacco); 0000-0001-8218-3178 (F. Ricca); 0000-0001-9672-3833 (G. Manco)

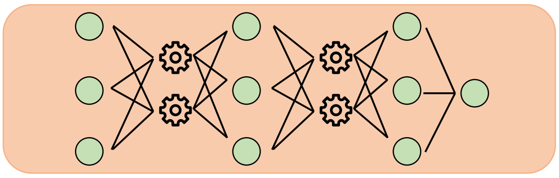


Figure 1: An example of Neuro[Symbolic] architecture

from the data (i.e., environment observations), and the aim is to define mathematical models based on them. Since the models are derived from the data, it is crucial to have a huge amount of them that are representative of the domain. Both approaches have limitations: the former has scalability and performance issues, while the latter lack of interpretability and partially removes human interaction. Hence, to exploit the potentiality of the two approaches, while limiting their weaknesses, we propose to employ hybrid approaches for improving existing predictive maintenance solutions. In this sense, after a critical review of current approaches, we devised a list of key advantages as a starting point for our research. In particular, to improve existing models, the novel ones should have (i) interpretability; (ii) robustness; and (iii) effectiveness properties. We believe these goals can be achieved by developing Neuro-symbolic approaches to predictive maintenance.

The paper is structured as follows. Section 2 explains the neuro-symbolic approach and its potentialities by describing some state-of-the-art models. Section 3 describes the context of predictive maintenance and its current approaches. In Section 4 we describe our proposal to exploit neuro-symbolic approaches for improving predictive maintenance existing approaches. Finally, we conclude the paper in Section 5.

2. Neuro-Symbolic approaches

Neuro-symbolic approaches are hybrid models exploiting both deductive (symbolic) and inductive (deep learning) approaches. Combining both approaches yield models that are more robust and accurate as well as explainable. Kautz [2] devised a taxonomy for neuro-symbolic integrations by dividing them according to their characteristics and how the combination is performed. The taxonomy establishes six categories: (*i*) **Symbolic Neuro Symbolic**; (*ii*) **Symbolic[Neuro]**; (*iii*) **Neuro|Symbolic** ; (*iv*) **Neuro:Symbolic->Neuro** ; (*v*) **Neuro_{Symbolic}**; (*vi*) **Neuro[Symbolic]**. Neuro-symbolic techniques are steadily growing interest in the research community, as well as being used in several areas. We then present an overview of state-of-the-art hybrid models to show their potentialities and their different usages according to the aforementioned taxonomy. Although in [2] six categories have been identified, to the best of our knowledge there are no works about **Neuro[Symbolic]**, i.e., methods in which neural networks exploit symbolic solvers inside their architecture, as shown in Figure 1.

Symbolic Neuro Symbolic Models in this category are mainly used in the context of Natural Language Processing (NLP), in which given a token (symbol), i.e., a word in a sentence, the aim is to generate its embedding for predicting the next tokens or for classifying them (sentiment analysis) or generating new ones. Architecture is shown in Figure 2. Mikolov et al. [3] propose

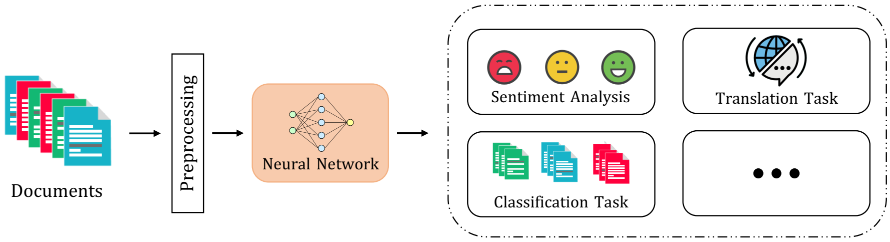
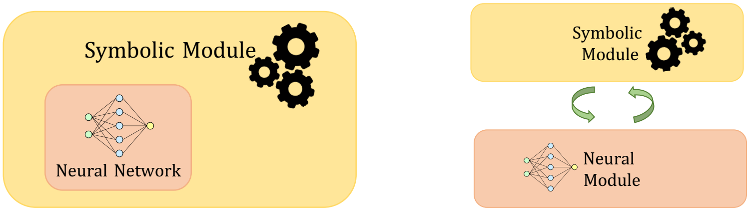


Figure 2: An example of Symbolic Neuro Symbolic architecture

word2vec, a neural architecture consisting of two sub-networks: CBOW and Skip-Gram. Unlike word2vec, GLOVE [4] uses matrix factorization for generating token embeddings. In addition, GLOVE optimizes the loss by combining the similarity of the words based on their occurrences.

Symbolic[Neuro] Models within this category exploit neural architecture as sub-networks invoked by an external symbolic solver. A classical example is AlphaGO [5], a framework that combines a Monte Carlo search tree strategy and Reinforcement learning models that learn how to evaluate game positions. The symbolic part, represented by the Monte Carlo strategy, uses the neural components to compute scores, associated with the nodes of the tree-based structure and, then, based on the scores continue the solution search. Garg et al. [6] propose SymNet whose aim is to plan the actions of an agent. SymNet is based on Relational Markov Decision Process (RMDP) to represent the knowledge and uses Dynamic Bayesian Network (DBN) for modeling RMDP. DBN is converted into a graph-based structure. Then, the embedding $\bar{v}$ and $\bar{s}$ are generated to represent nodes v and states s and fed into two *decoder layer* for producing $V(s)$ and $\pi(s)$, respectively. Action with the maximum probability in $\pi(s)$ is chosen. It is crucial to define methods for representing the knowledge base; for example, in [7], it is represented through sparse matrices.

Neuro|Symbolic Differently from the previous category (**Symbolic[Neuro]**) where neural networks are "sub-network", here models interact as equal in the global architecture. Figures 3a and 3b show the two categories. Models in **Neuro|Symbolic** are mainly employed for two tasks: planning and question-answering. *Planning*. Yang et al. [8] propose a unified framework, PEORL, that integrates symbolic planning with reinforcement learning for identifying which actions an agent will do in a given environment. An improved version is presented in [9] in which an intrinsic reward is introduced to optimize the reinforcement learning-based model. In PLANS [10], neural architecture is used for generating an action list starting from visual data. Then, the obtained outputs are fed into a rule-based solver that produces the sequences of final actions that an agent will do. In addition, a filtering system is used to keep only outputs for which the probability is above a given threshold. Symbolic Options for Reinforcement Learning (SORL) is proposed in [11] in which the authors assume that a function F maps the environment state in a set of symbolic states manipulated by a meta-controller and a symbolic planner for producing the actions that an agent will perform. *Question-answering*. [12] propose a hybrid model named Neural Symbolic Reader (NeRd), whose architecture is an encoder-decoder. The encoder creates embedding from the input question, while the decoder generates a symbolic program. The final answer is obtained by giving as input the program to a symbolic solver. [13] the authors presented a neuro-symbolic model for visual question answering. An important advantage of both [12, 13] is the interpretability of the results, since the symbolic program used for generating the final answers is self-explainable.



(a) (b)

Figure 3: An example of Symbolic[Neuro] (a) and Neuro|Symbolic (b) architecture

Neuro:Symbolic→Neuro This category includes all architectures in which the knowledge base is processed by a neural model. Figure 4a shows the architecture of such an approach. In [14], the authors propose a language model, named Neural-Symbolic Language Model whose aim is to improve the inductive bias. Neural Markov Logic Networks (NMLN) is developed in [15]. Markov Logic Networks are probabilistic models in which the logic is used to represent data and statistics for prediction tasks. NMLN exploits neural networks for estimating probability distributions that govern the logic rules through min-max entropy. In [16], a framework, named KRISP, combining symbolic (explicit) and implicit knowledge is proposed. Given an image as input, the model extracts symbolic information, called visual concepts, that are then combined with a knowledge-graph (KG). A Graph Neural Network and Transformer architecture are used to estimate the probability distributions of answers.

Neuro_{Symbolic} Systems that integrate logic rules into neural network weights are included in this category; an architectural example is shown in Figure 4b. In [17] the authors propose Logic Tensor Network (LTN) whose aim is to find a way to differentiate logic rules based on first-order logic formalism. To differentiate logic operators, authors consider the method proposed in [18] in which differentiable operations are defined instead of using logic operations. Hoernle et al. [19] propose Multiplexnet that integrates logic constraints into the neural network computation to guide its training. Assuming that the rules are in Disjunctive Normal Form (DNF), the goal is to find a data transformation that makes the DNF satisfactory. To do this, the activation functions are modified and a component representing the grade of violation of the constraints is added to the loss. [20] integrate hard constraint into the neural network. Neural Logic Machines (NLM) [21] combines inductive and deductive approaches. The grounding of the predicate is converted into boolean tensors that are manipulated by exploiting differentiable operators defined *meta-rule*. In [22] the authors develop SATNet to address the MAXSAT problems by using a neural network.

3. Predictive Maintenance

Maintenance policies are fundamental within industries, whose aims are manifold: anticipating failures, reducing costs associated with unplanned downtime, and methodologies for systems restoring. They provide data-driven insights that allow organizations to proactively manage their

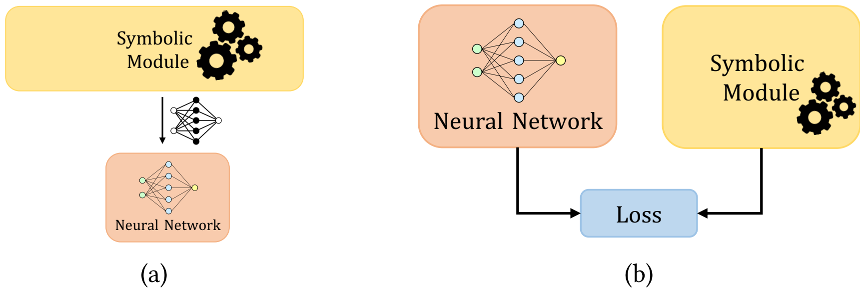


Figure 4: An example of Neuro:Symbolic $\rightarrow$ Neuro (a) and Neuro_{Symbolic} (b) architecture

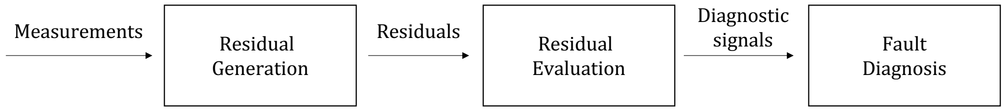


Figure 5: Overall mechanism of model-based approach

equipment to manage such issues. In particular, predictive maintenance aims to continuously monitor the data provided by sensors installed into the system in order to identify failure and take solutions to restore the operational status. Predictive maintenance can be summarized in three key steps: (1) monitoring systems' parameters; (2) setting parameter thresholds for identifying anomalies; (3) defining methods and tools to recover systems functionalities. Methods and tools for predictive maintenance are constantly evolving, in particular, state-of-the-art methods exploit automatic algorithms for (*i*) preprocessing sensor data, and (*ii*) evaluating the systems' correct functioning. In the area of predictive maintenance, the common algorithmic approaches are: *Model-based* and *Data-driven*.

Model-based. Model-based approaches use mathematical models for identifying failures, they exploit a knowledge base given by experts within a specific topic. In particular, the overall mechanism of model-based approaches can be depicted in three sequential steps, as shown in Figure 5. Given a component X, a timestamp t, a parameter x_t describing the regular performance level of x, and a second parameter $\widehat{x}_t$ representing the real performance level, we can measure the residuals by computing $res = x_t - \widehat{x}_t$. Then, res is passed through a set of logic rules (defined by experts) able to identify anomalies. The final stage, Fault Diagnosis, utilizes the output from the preceding step to obtain additional data in order to examine the present malfunction, and proactively mitigate any prospective irregularities. Human knowledge may be represented as symbols by formal languages such as First Order Logic (FOL) or Propositional Logic. Symbolic systems exploit a deductive approach: starting from a general knowledge base (KB) within a specific topic, it is possible to infer new knowledge.

Data-driven. Inductive approaches (such as data-driven) perform bottom-up strategies: starting from observations, they realize models able to generalize the observation for a larger population, as well as infer new data. Among data-driven approaches, artificial intelligence has been exploited for its capability to create models able to analyze large datasets in order to identify anomalous patterns. In predictive maintenance, the concept of "anomalous patterns" usually coincides with the concept of "systems inefficiency", hence, the rising of possible failures. Various kinds of observations are used, such as data coming from sensors (temperature, humidity, speed), and logic data. To develop efficient data-driven systems, a huge amount of data is needed, as well as massive computing power.

4. Weaknesses of existing methods and our proposal

Predictive maintenance current approaches (model-based and data-driven) have several limitations, hence, we provide an overview of the main weaknesses. To overcome those issues, we propose to create predictive maintenance solutions exploiting hybrid techniques based on three main key points. Moreover, we present a possible use-case scenario.

4.1. Limitations of current approaches

Data-driven approach yields more high-quality outcomes. Model-based approaches offer a straightforward interpretation due to the fact that the parameters responding to physical phenomena within systems correspond with behavioral models. Nonetheless, generating accurate models proves difficult in practice, particularly when facing complex systems in which various types of physical phenomena take place. Even when such a model exists, it generally represents a particular physical phenomenon that was generated during specific experimental conditions. In consequence, conducting experiments for different operating conditions can be expensive, limiting the potential use of this approach. However, the widespread availability of sensors and increased computing power has facilitated the use of artificial intelligence techniques, leading to the proliferation of data-driven methods. These methods utilize artificial intelligence tools to transform monitoring data into behavioral models. Data-driven approaches provide a trade-off in terms of complexity, cost, accuracy, and applicability. When compared to model-based approaches, data-driven approaches are suitable for systems in which obtaining monitoring data that represents degradation behavior proves accessible. Yet, one of the limitations of data-driven methods is the potentially long learning time required. In terms of accuracy, data-based methods offer less precise results than model-based methods, but they are less complex and therefore more applicable. Here we provide a summary of the aforementioned issues.

- **Model-based**: (*i*) experts can make assumptions that may not always reflect the complexity of the real-world problems they are trying to solve; (*ii*) specific and deep knowledge can lead to high costs for industries; (*iii*) poor results;
- **Data-driven**: (*i*) real data can be noisy, inconsistent, and sparse, this can lead models to overfit or to develop biases; (*ii*) black-box data-driven models are not explainable; (*iii*) historical data could not be fully representative of real-world scenarios.

4.2. Neuro-symbolic advantages

Interpretability Interpretability is the main characteristic of the deductive approaches as in such models it is easy to understand the reasoning behind predictions, while the black-box nature of neural networks and data-driven approaches make them less interpretable, i.e., other techniques are needed to interpret the generated outputs. Several application scenarios, such as bioinformatics, robotics, and so on, need effective and interpretable models. For instance, when a disease is predicted, it is crucial having an accurate prediction (effectiveness) as well as understand which factors led to that prediction (interpretability). Therefore, it is desirable to use hybrid approaches, i.e., neuro-symbolic methods, that combine the power of neural networks and the interpretability of logic formalisms (deductive approaches) [12, 13]. Within the explainability area, the most common representations of the knowledge are (*i*) Knowledge Graphs (KGs), i.e., graph-based structure, and (*ii*) Tree, i.e., tree-based structure. For instance, in [23] a neuro-symbolic model is used for stock prediction. The authors utilize a knowledge graph for representing relationships among financial events. The proposed strategy can be positioned within the **Neuro:Symbolic->Neuro** categorization (see Section 2), it allows for improving the performance in stock prediction tasks as well as providing reasons for price variations through the generation of new nodes. Tree-based structures are employed in techniques such as counterfactuality and surrogate models.

Performance Within model-based approaches, symbolic solvers and reasoning techniques are widely exploited. They can derive new data given a starting knowledge base, while neural networks often depend on training data, i.e., they provide high performances on data that are similar to the training examples, but when dealing with different and more complex data, the performance decrease dramatically. This problem is widely known within the model-based approaches Integrating knowledge derived from symbolic models into neural network architectures can increase performances. In this sense, such integration could be exploited for (*i*) guiding the neural network training, (*ii*) for filtering its output through logic constraints, and (*iii*) for improving performance by integrating symbolic-based data structures (such as knowledge graphs) with the observations.

Robustness Training data for data-driven models may be incomplete and not fully representative. This can lead systems to be susceptible to input data, i.e., varying a single pixel within an image can completely change the result. Narrow intelligence and Pointillistic intelligence [24] are both artificial intelligence categories, where models lack robustness from different viewpoints. In Narrow Intelligence models are devised for specific tasks, i.e., computer vision applications or natural language processing. Although these models exhibit high performances within their respective categories, their effectiveness may drastically decrease when applied to similar contexts. Additionally, adapting these models for transfer learning methodologies can be a challenging endeavor. In Pointillistic Intelligence models show great results but may fail in unpredictable cases. Lack of generalization and flexibility can lead to such issues. Neuro-symbolic approaches integrate human expertise and real-world observations in order to accurately create robust systems, overcoming problems related to noise within data and bias produced by human assumptions on complex real-world systems.

4.3. Use-cases

Shy approaches for hybridizing techniques have been provided [25, 26]. In particular, within the root cause analysis area, i.e., a sub-area of predictive maintenance whose aim is to analyze and study root causes of faults or problems, Abele et al. [25] proposed a neuro-symbolic architecture for alarm flooding problem, which can be positioned in the **Neuro|Symbolic** category (see Section 2). Data are represented by using a Bayesian Network (BA): specifically, a directed acyclic graph (DAG) $G = (V, E)$ is employed in which V represents the set of vertices/nodes and E is the edge set. Each node $v_i \in V, i = 1..., N$ corresponds to a random variable in BA, while the edges represent the conditional dependency among nodes. *Prolog*, a rule engine, is used to infer the relationships among entities, and for creating G. The stages of the neuro-symbolic process are: (*i*) starting from knowledge, the graph is generated; (*ii*) exploiting active learning, the initial dataset is augmented; (*iii*) a Bayesian Classifier is trained on the augmented dataset; (*iv*) machine learning-based model is used to assess the graph generated at the beginning. Moreover, we further propose a generic use-case by exploiting the **Symbolic[Neuro]** strategy (see Section 2). The aim is to monitor the operational status of a train. Suppose to have a failure, i.e., a carriage door is blocked. Given an ontology, describing all components of the train, an expert define the relationships among components by using a logical formalism, modeling them as a tree-based structure. Thanks to the neuro-symbolic approach, it is possible to define a "formal" representation of the relationships among components. In addition, a machine/deep learning-based model, such as an auto-encoder, could be used for anomaly detection in each component. In this context, a classifier C is used for identifying anomalous behaviors, based on historical data of a given component, and a tree-based structure A represents the relationships among the components. The proposed approach allows (*i*) recognizing anomalies within single components exploiting C; (*ii*) to identify which causes have determined the failure through A and (*iii*) explain the whole process of root cause analysis.

5. Conclusion

The neuro-symbolic approaches show promise in the field of predictive maintenance. We aim to further explore and develop these techniques, which have the potential to overcome some of the limitations of the traditional model-based and data-driven approaches. The final goal is to effectively deliver novel neuro-symbolic models for predictive maintenance so that one can leverage new architectures that prioritize interpretability, robustness, and maintain high-performance levels.

Acknowledgments

This work was partially supported by the Italian Ministry of Industrial Development (MISE) under project MAP4ID "Multipurpose Analytics Platform 4 Industrial Data", N. F/190138/01-03/X44 and by Italian Ministry of Research (MUR) under PNRR project PE0000013-FAIR, Spoke 9 - Green-aware AI – WP9.1." and by the Departmental Strategic Plan (PSD) of the University of Udine—Interdepartmental Project on Artificial Intelligence (2020-25).

References

[1] T. Zonta, C. A. da Costa, R. da Rosa Righi, M. J. de Lima, E. S. da Trindade, G. P. Li, Predictive maintenance in the industry 4.0: A systematic literature review, Computers & Industrial Engineering 150 (2020) 106889.

[2] H. Kautz, The third ai summer: Aaai robert s. engelmore memorial lecture, AI Magazine 43 (2022) 105–125.

[3] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, in: ICLR (Workshop Poster), 2013.

[4] J. Pennington, R. Socher, C. D. Manning, Glove: Global vectors for word representation, in: EMNLP, ACL, 2014, pp. 1532–1543.

[5] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. P. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, D. Hassabis, Mastering the game of go with deep neural networks and tree search, Nat. 529 (2016) 484–489.

[6] S. Garg, A. Bajpai, Mausam, Symbolic network: Generalized neural policies for relational mdps, in: ICML, volume 119 of *Proceedings of Machine Learning Research*, PMLR, 2020, pp. 3397–3407.

[7] W. W. Cohen, H. Sun, R. A. Hofer, M. Siegler, Scalable neural methods for reasoning with a symbolic knowledge base, in: ICLR, 2020.

[8] F. Yang, D. Lyu, B. Liu, S. Gustafson, PEORL: integrating symbolic planning and hierarchical reinforcement learning for robust decision-making, in: IJCAI, ijcai.org, 2018, pp. 4860–4866.

[9] D. Lyu, F. Yang, B. Liu, S. Gustafson, SDRL: interpretable and data-efficient deep reinforcement learning leveraging symbolic planning, in: AAAI, AAAI Press, 2019, pp. 2970–2977.

[10] R. Dang-Nhu, PLANS: neuro-symbolic program learning from videos, in: NeurIPS, 2020.

[11] M. Jin, Z. Ma, K. Jin, H. H. Zhuo, C. Chen, C. Yu, Creativity of AI: automatic symbolic option discovery for facilitating deep reinforcement learning, in: AAAI, AAAI Press, 2022, pp. 7042–7050.

[12] X. Chen, C. Liang, A. W. Yu, D. Zhou, D. Song, Q. V. Le, Neural symbolic reader: Scalable integration of distributed and symbolic representations for reading comprehension, in: ICLR, 2020.

[13] K. Yi, J. Wu, C. Gan, A. Torralba, P. Kohli, J. Tenenbaum, Neural-symbolic VQA: disentangling reasoning from vision and language understanding, in: NeurIPS, 2018, pp. 1039–1050.

[14] D. Demeter, D. Downey, Just add functions: A neural-symbolic language model, in: AAAI, AAAI Press, 2020, pp. 7634–7642.

[15] G. Marra, O. Kuzelka, Neural markov logic networks, in: UAI, volume 161 of *Proceedings of Machine Learning Research*, AUAI Press, 2021, pp. 908–917.

[16] K. Marino, X. Chen, D. Parikh, A. Gupta, M. Rohrbach, KRISP: integrating implicit and symbolic knowledge for open-domain knowledge-based VQA, in: CVPR, Computer Vision Foundation / IEEE, 2021, pp. 14111–14121.

[17] S. Badreddine, A. S. d'Avila Garcez, L. Serafini, M. Spranger, Logic tensor networks, Artif.

Intell. 303 (2022) 103649.
[18] E. van Krieken, E. Acar, F. van Harmelen, Analyzing differentiable fuzzy logic operators, Artif. Intell. 302 (2022) 103602.
[19] N. Hoernle, R. Karampatsis, V. Belle, K. Gal, Multiplexnet: Towards fully satisfied logical constraints in neural networks, in: AAAI, AAAI Press, 2022, pp. 5700–5709.
[20] E. Giunchiglia, T. Lukasiewicz, Multi-label classification neural networks with hard logical constraints, J. Artif. Intell. Res. 72 (2021) 759–818.
[21] H. Dong, J. Mao, T. Lin, C. Wang, L. Li, D. Zhou, Neural logic machines, in: ICLR (Poster), OpenReview.net, 2019.
[22] P. Wang, P. L. Donti, B. Wilder, J. Z. Kolter, Satnet: Bridging deep learning and logical reasoning using a differentiable satisfiability solver, in: ICML, volume 97 of *Proceedings of Machine Learning Research*, PMLR, 2019, pp. 6545–6554.
[23] S. Deng, N. Zhang, W. Zhang, J. Chen, J. Z. Pan, H. Chen, Knowledge-driven stock trend prediction and explanation via temporal convolutional network, in: WWW (Companion Volume), ACM, 2019, pp. 678–685.
[24] G. Marcus, The next decade in AI: four steps towards robust artificial intelligence, CoRR abs/2002.06177 (2020).
[25] L. Abele, M. Anic, T. Gutmann, J. Folmer, M. Kleinsteuber, B. Vogel-Heuser, Combining knowledge modeling and machine learning for alarm root cause analysis, in: MIM, International Federation of Automatic Control, 2013, pp. 1843–1848.
[26] M. Saez, F. P. Maturana, K. Barton, D. M. Tilbury, Anomaly detection and productivity analysis for cyber-physical systems in manufacturing, in: 13th IEEE Conference on Automation Science and Engineering, CASE 2017, Xi’an, China, August 20-23, 2017, IEEE, 2017, pp. 23–29.