Neurosymbolic AI for Explainable Recommendations in Frontend UI Design - Bridging the Gap between Data-Driven and Rule-Based Approaches

Sunil Raj Thota[1], Saransh Arora[2]

[1]*Independent Researcher/Sr. Software Engineer, AI/ML & Full-stack Web Dev, Boston, MA, USA*
[2]*Independent Researcher/Sr. Data Engineer, AI/ML & Data Engineering, Seattle, WA, USA*

---***---

Abstract- This paper proposes a novel approach utilizing neurosymbolic artificial intelligence (AI) techniques to enhance the interpretability and effectiveness of recommendations in front-end user interface (UI) design. By integrating both data-driven and rule-based methodologies, our framework aims to bridge the gap between conventional recommendation systems and human-understandable decision-making processes. We leverage neurosymbolic AI to combine statistical learning from large-scale data with symbolic reasoning capabilities, enabling transparent and interpretable recommendations that align with user preferences and design principles. Through a series of experiments and case studies, we demonstrate the efficacy of our approach in providing explainable recommendations for front-end UI design tasks, facilitating more intuitive and user-centric interfaces.

Keywords-Neurosymbolic, AI, Data-Driven Approaches, Rule-Based Approaches Bridging the Gap Interpretability

I INTRODUCTION

In recent years, there has been growing interest in developing recommendation systems for frontend UI design that can effectively bridge the gap between data-driven approaches and rule-based methods. While data-driven techniques, such as collaborative filtering and matrix factorization, excel at capturing complex patterns in user preferences, they often lack transparency and interpretability. On the other hand, rule-based approaches offer explicit control over recommendation logic but may struggle to handle the vast amount of data and evolving user behaviors [1].

Researchers have resorted to neurosymbolic artificial intelligence, a novel technique that successfully blends the advantages of neural networks with symbolic reasoning, in order to address these issues. By integrating deep learning models with symbolic knowledge representations, neurosymbolic AI offers a promising framework for developing recommendation systems that are both accurate and interpretable. Furthermore, the incorporation of knowledge graphs provides a structured representation of domain-specific information, enabling more effective reasoning and decision-making.

In this paper, we propose to explore the application of neurosymbolic AI for explainable recommendations in frontend UI design, with a particular focus on leveraging knowledge graphs. We aim to develop a recommendation framework that can seamlessly integrate data-driven insights with domain knowledge encoded in the form of a knowledge graph. In doing so, we aim to improve the transparency, interpretability, and effectiveness of recommendation systems for frontend UI design.

In today's world, artificial intelligence (AI) has garnered widespread interest across a variety of application sectors, including the business sector. [2] Predictive maintenance, in particular, plays a significant role because it enables businesses to avoid internal system failures in a preventative manner and reduces the costs associated with business interruptions. People are now using techniques based on models and data to develop design, optimization, diagnostic, and maintenance stages. Model-based strategies utilize mathematical models, along with background information from human specialists, to achieve their goals. We utilize mathematical models to describe the interactions that govern a certain environment. On the other hand, data-driven approaches are inductive methods. These approaches involve the creation of models by generalizing from the data (that is, observations of the environment), with the objective of defining mathematical models based on the insights gained from the data. Since the models originate from the data, it is crucial to have a significant number of models that accurately reflect the region. The first method has problems with scalability and performance, while the second method is not interpretable and eliminates human engagement to some extent. Both methods have their drawbacks. Therefore, in order to make the most of the potential offered by both approaches while simultaneously minimizing their drawbacks, we advocate the use of hybrid approaches to enhance the predictive maintenance solutions that are now in place.

Having analyzed the various approaches currently in use, we compiled a list of the most significant benefits to serve as a foundation for our investigation. It is also important that the novel models have the following characteristics: (i) interpretability; (ii) resilience; and (iii) efficacy features. This will enable the enhancement of the current models. We are of the opinion that these objectives can be accomplished through the development of neuro-symbolic methods for predictive maintenance practices.

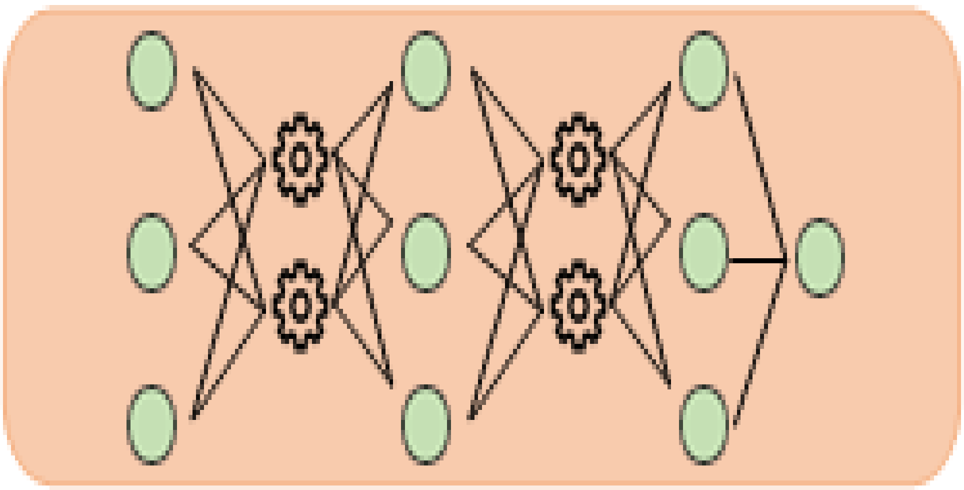

Figure 1: Neuro [Symbolic] architecture

Neurosymbolic architecture refers to a computational framework that integrates neural network-based learning with symbolic reasoning methods. This hybrid approach combines the ability of neural networks to learn patterns and representations from data with the logical reasoning capabilities of symbolic systems.

Neural networks typically perform tasks like feature extraction, pattern recognition, and representation learning from large-scale data in neurosymbolic architecture. These neural components learn to encode complex relationships and patterns in the data, capturing its underlying structure.

The structure of the paper is as described below.

- Discussing various models that are considered to be state-of-the-art, and provides an explanation of the neuro-symbolic approach and its potentialities.
- The context of predictive maintenance and the ways that are currently being used are discussed.
- Provides a description of our proposal to utilize neuro-symbolic approaches for the purpose of enhancing the predictive maintenance methods that are now in use.

Through this research, we hope to contribute to the advancement of recommendation systems in UI design and provide valuable insights into the potential of neurosymbolic AI for reasoning over rule-based design and knowledge graphs. Ultimately, our goal is to empower designers and developers with tools that can generate personalized and insightful recommendations while maintaining transparency and interpretability in the decision-making process.

The study aims to provide a rule-based design process for neurosymbolic systems, emphasizing the integration of knowledge-based systems (KBS) and deep learning models (DL). The study integrates both theoretical and practical integration components, making it applicable to a wide range of model combinations and application scenarios. This design process addresses the complementary features of symbolic and subsymbolic paradigms, aiming to harness their combined benefits. However, the lack of explicit design rules for neurosymbolic models necessitates an examination of the motivations for integration alongside contextual and practical considerations, such as available resources, required data types, and the scope of work. While neurosymbolic design methods in existing literature lay a foundational understanding and outline desired system characteristics, they fall short in offering specific implementation guidelines. Consequently, there is a notable absence of clear instantiation examples highlighting integration perspectives. [3]

Furthermore, we offer specialized integration templates to provide a more detailed characterization of neurosymbolic systems. When considering neurosymbolic techniques from the perspective of integration, we can distinguish two primary categories: introduction (also known as insertion) and extraction. When compared to extraction approaches, insertion approaches include the integration of many models within a single framework. On the other hand, extraction approaches involve mining one of the models from another. The provided description and the reviewed literature suggest three possible integrations: KBS insertion into DL models, DL insertion into KBS, and KBS extraction from DL models. This document outlines the parameters for each of these three integration scenarios, providing a template to represent those parameters. We offer an instantiation example for each integration template to further demonstrate the applicability of the given template-based design process. Each of these templates is associated with a different interaction that could occur. These templates provide a description of neurosymbolic systems that have actually existed in the past, demonstrating how the proposed method can accurately profile these frameworks.

II NEUROSYMBOLIC SYSTEM DESIGN

Cutting-edge technology A set of methodologies aimed at integrating antagonistic AI models is referred to as neurosymbolic (or hybrid) artificial intelligence. These methods aim to develop innovative approaches that amalgamate the most effective elements of symbolic and subsymbolic methods into a unified approach. [4] proposed a theory regarding the potential advantages that could be gained by integrating symbolic and subsymbolic systems, which for the purposes of this discussion, are represented by rule-based systems and neural networks, respectively. "The ability of neural networks to perform tasks that would otherwise prove difficult or intractable to symbolic computing systems is now recognized," [5] states. "This ability has been recognized now." Neurosymbolic systems assert that they integrate the benefits of symbolic and subsymbolic models, enabling the resulting hybrid system to tackle problems beyond the scope of a single model. This statement explains the theory underlying neurosymbolic systems. You can find a more recent discussion of this assertion in [6], which offers an overview of the characteristics of symbolic and subsymbolic models. The research findings summarize the fundamental concept of neurosymbolic integration as "the creation of a system that combines the benefits of both methods: the ability to learn from the environment and the ability to reason about the results." One of the first ways that properties that these systems should have were defined was by the work of McGarry and coworkers. The study suggests that the following should be some of the most important parts of a neurosymbolic system: (i) the ability to think clearly even when the data is noisy or incomplete; (ii) the capacity to learn little by little from new experiences; and (iii) the capacity to generalize and explain the models' line of thought. A recent study [8] has helped us narrow down the best qualities of neurosymbolic systems to two main ones..

Neuro Symbolic AI

Neuro-symbolic artificial intelligence is an interdisciplinary field in computer science that combines neural networks, which are a component of deep learning, with symbolic reasoning techniques. By combining the positive aspects of both approaches, it seeks to close the gap that exists between statistical learning and symbolic thinking. In addition to utilizing the tremendous pattern recognition capabilities of neural networks, this hybrid approach enables robots to reason symbolically about their own actions. A more sophisticated form of artificial intelligence than its conventional counterpart, it employs deep learning neural network topologies and combines them with symbolic reasoning techniques. With neural networks, for example, we have been able to discern the type of shape or color that an item possesses. On the other hand, it is possible to push it further by employing symbolic thinking in order to expose more exciting properties of the item, such as its area, volume, and related characteristics.

III RULE-BASED APPROACHES IN NEUROSYMBOLIC

Neurosymbolic AI for reasoning that is based on rules combines symbolic rules with neural network architectures to make reasoning more effective and easier to understand.

In traditional rule-based systems, logical rules are manually defined to represent knowledge and infer new information based on logical deductions. However, these systems may struggle with handling uncertainty, noisy data, and complex patterns in real-world scenarios. On the other hand, neural networks excel at learning patterns and representations from large-scale data but often lack transparency and interpretability.

Neurosymbolic AI aims to combine the strengths of both symbolic and neural approaches to address these limitations. In the context of reasoning, this means adding symbolic rules to neural network models so they can do reasoning tasks while still using neural networks' data-driven learning features.

There are several ways to incorporate rule-based reasoning into neural architectures:

Neural-Symbolic Integration:

This approach involves explicitly encoding symbolic rules as part of the neural network architecture. For instance, we can represent logical rules as trainable parameters or embed them within the network structure itself. This allows the neural network to learn to reason based on both data and predefined rules.

Neural-Symbolic Fusion:

This approach combines neural embeddings learned from data with symbolic representations to perform reasoning tasks. This could involve learning embeddings for entities and relations in a knowledge graph, then using symbolic rules to perform logical inference over these embeddings.

Differentiable Rule Learning:

This approach involves learning symbolic rules in a differentiable manner within a neural network framework. This lets the network improve both the neural parameters and the symbolic rules at the same time based on a goal function. This combines data-driven learning with symbolic reasoning in a useful way.

Neurosymbolic Execution: This approach involves executing symbolic rules directly within neural networks. Instead of relying on separate rule engines or inference mechanisms, neural networks are augmented with the ability to perform logical operations and symbolic reasoning as part of their computation graph.

Need for Neuro Symbolic AI

It is evident that symbols are an essential component of communication, which helps the intelligence of people when one takes into consideration how people think and reason. Researchers attempted to imitate robot symbols in order to make them function in a manner that is comparable to that of humans. The explicit incorporation of human knowledge and behavioral standards into computer programs was necessary for rule-based symbolic artificial intelligence design. The addition of more rules not only increased the cost of the systems, but also decreased their accuracy.

Researchers studied a technique that was more data-driven to address these issues, which sparked the attraction of neural networks. Despite the constant information input required for symbolic artificial intelligence, a sufficiently large dataset can enable neural networks to train themselves. Despite everything operating smoothly, the difficulty in comprehending the model and the volume of data necessary for continued learning necessitate the implementation of a more appropriate system.

In spite of its effectiveness in large-scale pattern recognition, deep learning is not able to identify compositional and causal structures from data. Despite symbolic models' design to capture intricate linkages, they effectively capture compositional and causal patterns.

The drawbacks of these two approaches led to their merging to create neuro-symbolic artificial intelligence, which outperforms each when used separately. The goal is to achieve a higher level of system intelligence by combining learning with logic. The integration of domain knowledge and common-sense thinking that is given by symbolic AI systems is anticipated to be beneficial to deep learning, as stated by researchers. For instance, a neuro-symbolic system uses the logic of symbolic artificial intelligence to detect shapes more accurately, and a neural network's pattern recognition capabilities to identify items.A neuro-symbolic system works by employing logical reasoning and language processing in order to provide a response to the question in the same manner that a human would. It is, nevertheless, more effective than neural networks and requires a significantly smaller amount of training data than neural networks do.

Neuro-Symbolic approaches

Neuro-symbolic approaches are hybrid models that combine both inductive (deep learning) and deductive (symbolic) approaches. Together, these two methods provide models that are not only more resilient and accurate but also more easily explicable. [9] developed a taxonomy for neuro-symbolic integrations by categorizing them based on the features they possess and the method of completion.

Neuro-symbolic techniques are gaining interest in the research community and finding application in various fields. After that, we provide an overview of the most advanced hybrid models, demonstrating their potential and the various applications that they can be used for in accordance with the taxonomy that was previously described. Even though six categories were identified in [10], as far as we know, no works have been written about neuro [symbolic]. This refers to methods in which neural networks use symbolic solvers in their design, as shown in Figure 1.

Symbolic Neuro Symbolic

The majority of the time, models that fall into this category are utilized in the context of natural language processing (NLP). In this context, the objective is to construct an embedding of a token, which is a symbol, such as a word in a sentence. We did this with the intention of predicting subsequent tokens, classifying them (through sentiment analysis), or producing new tokens. Figure 2 shows a representation of the architecture. It was proposed by [11].

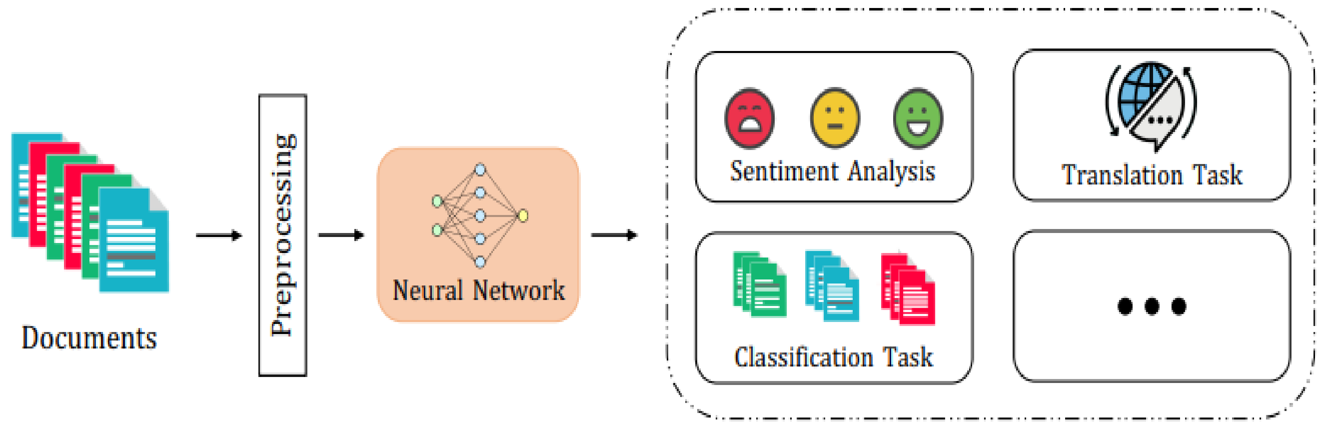

Figure 2: symbolic neurosymbolic architecture

The majority of the time, models that fall into this category are utilized in the context of NLP. In this context, the objective is to construct an embedding of a token, which is a symbol, such as a word in a sentence. We did this with the intention of predicting subsequent tokens, classifying them (through sentiment analysis), or producing new tokens. Figure 2 shows a representation of the architecture. It is proposed by. [12].

Neuro Symbolic In contrast to the preceding category (Symbolic [Neuro]), in which neural networks are considered to be "sub-networks," the models in this category interact with one another in an equal manner inside the global architecture. Figures 3a and 3b depict two different groups. When it comes to neuro-symbolic, the most common applications for models are in the areas of query response and planning. Making plans. [13] proposed a unified framework, PEORL, to determine the behaviors an agent would perform in a specific environment. This framework mixes symbolic planning with reinforcement learning. In [14], they describe an updated version that incorporates an intrinsic reward to enhance the optimization of the reinforcement learning model. PLANS [15] utilizes neural architecture to construct an action list, starting with visual data. Next, a rule-based solver receives the obtained outputs and generates the sequences of final actions that an agent will execute. Additionally, a filtering method selects and retains only those outputs that exceed a specific threshold in likelihood. The authors of [16] present a technique known as Symbolic Options for Reinforcement Learning (SORL).

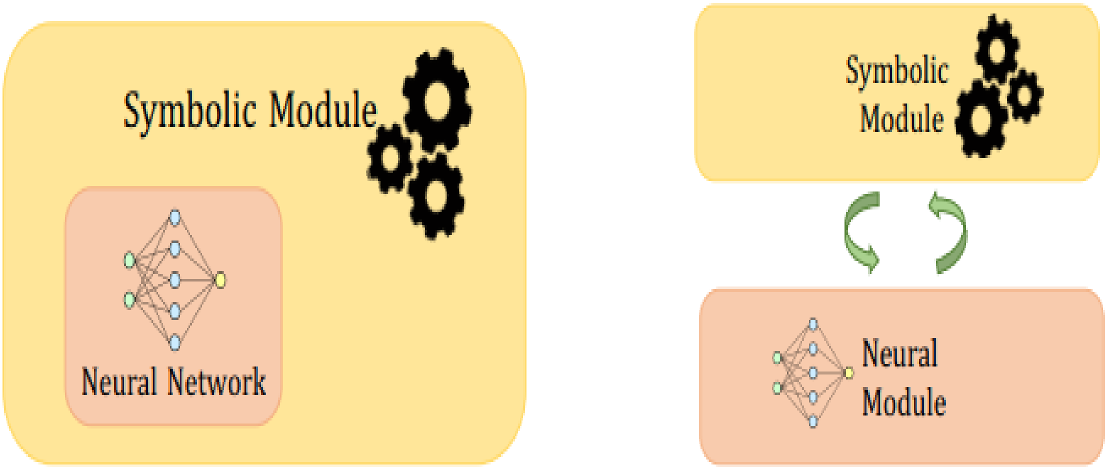

Figure 3: a symbolic architecture with neuro components,

Neuro:Symbolic Neuro - This category includes any and all architectures that utilize a neural model for the processing of the knowledge base. The architecture of such an approach is depicted in Figure 4a, respectively. The authors of [17] present a language model, which they call the Neural-Symbolic Language Model. The purpose of this model is to improve the inductive bias. The development of neural Markov Logic Networks (NMLN) may be found in. Probabilistic models, known as Markov Logic Networks, are characterized by the use of logic to describe data and statistics in order to solve prediction tasks. NMLN employs neural networks to estimate probability distributions that govern the logic rules using min-max entropy. In [18], we propose a framework that integrates symbolic (explicit) and implicit knowledge. This framework is provided by the term KRISP. Upon receiving an image as input, the model extracts symbolic information known as visual concepts. We then couple these concepts with a knowledge graph (KG). The estimation of the probability distributions of the replies is accomplished through the utilization of graph neural networks and transformer architecture.

Neuro {Symbolic} Figure 4b depicts one example of an architectural design that falls within this category. This category includes systems that incorporate logic principles into the weights of neural networks. On the basis of first-order logic formalism, the authors of [19] propose a logic tensor network (LTN) with the intention of locating a method that can discriminate logic rules. The authors consider [20's] strategy, which defines differentiable operations instead of employing logic operations to distinguish logic operators. Proposed by [21], Multiplexnet is a neural network optimization technique that integrates logic constraints into neural network computation to guide its training. The objective is to locate a data transformation that, given the assumption that the rules are in disjunctive normal form (DNF), will result in the DNF being sufficient. To achieve this, we add a component representing the degree of constraint violation to the loss and adjust the activation functions accordingly. Hard limitations should be incorporated into the neural network under consideration. The techniques of inductive reasoning and deductive reasoning are combined in neural logic machines (NLM). Through the utilization of a differentiable operator's defined meta-rule, the grounding of the predicate is transformed into boolean tensors, which may then be employed for manipulation. The authors of [22] created SATNet to solve the MAXSAT issues by utilizing a neural network.

Predictive Maintenance

The rules that govern maintenance are essential in industries, and their objectives are diverse. These include preventing failures, reducing costs associated with unplanned downtime, and developing methods for recovering systems. They provide organizations with data-driven insights that enable them to monitor and manage their equipment in a proactive manner in order to address such difficulties. Predictive maintenance specifically aims to continuously monitor the data from inserted sensors in the system. We do this to identify failures and implement solutions to restore the equipment's operational status. Specifically, the goal of predictive maintenance is to continuously monitor the data provided by sensors inserted into the system. We do this to identify failures and implement solutions to restore the system's functionality.

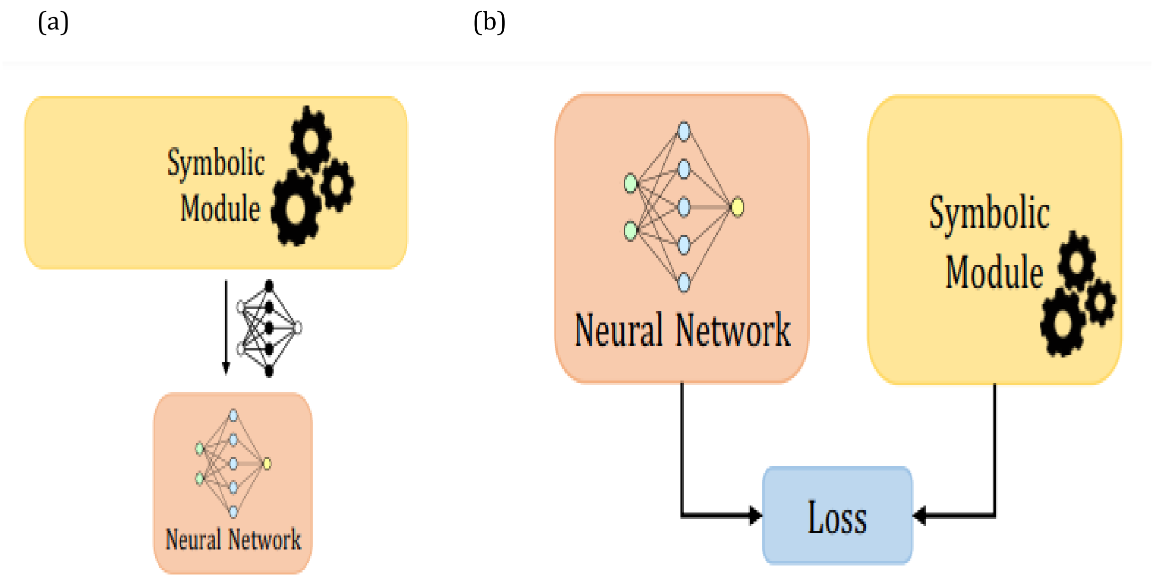

Figure 4: Neuro: Symbolic → Neuro (a) and Neuro_{Symbolic} (b) architecture

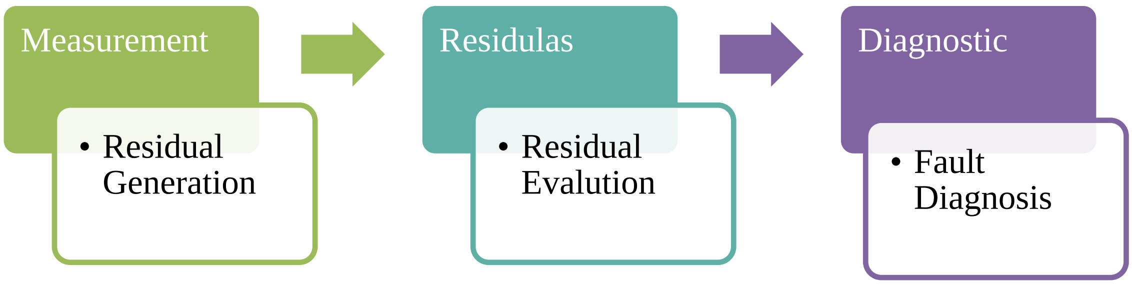

Figure 5: "The model-based approach encompasses the overall mechanism."

Predictive maintenance can be broken down into three primary steps, which are as follows: (3) designing methods and tools to recover system functionalities; (4) monitoring the parameters of the systems; (5) establishing parameter thresholds for the purpose of discovering abnormalities. Specifically, state-of-the-art methods utilize automatic algorithms for (i) preprocessing sensor data and (ii) evaluating the correct working of the systems. Methods and tools for predictive maintenance are continually improving, and this is especially true for automatic algorithms. Model-based and data-driven algorithms are the two most popular techniques utilized in the field of predictive maintenance.

Model-based. Methods that are based on models make use of mathematical models in order to discover failures; these methods also make use of a knowledge base that is provided by specialists in a particular field. Figure 5 illustrates the general mechanism of model-based techniques in three successive steps. In particular, this is the case. In the event that we are provided with a component X, a timestamp, a parameter that describes the usual performance levef, and a second parameter that represents the actual performance level, we are able to determine the residuals by computing $e = -$ Specialists then put the data through a series of logic rules to identify irregularities. . In the final stage, which is called fault diagnosis is, the output from the step before it is used to obtain further data in order to investigate the current failure and take preventative measures against any potential anomalies. Formal languages such as First Order Logic (FOL) an position logical are examples of possible ways in which human knowledge can be expressed as symbols Symbolic systems employ a deductive method, enabling the inference of new knowledge from a generic knowledge base (KB) relevant to a specific subject.

Data-driven. Bottom-up tactics are carried out using inductive procedures, such as data-driven approaches. These approaches begin with observations and then develop models that are able to generalize the observation to a broader population and infer additional data. As a data-driven method, artificial intelligence has been used because it can be used to make models that can look at very large datasets and find patterns that aren't normal. "Anomalous patterns" are often related to the idea of "systems inefficiency" in predictive maintenance, which means that there are more chances for things to break. It uses a variety of information, such as logic data and data from sensors (like temperature, humidity, and speed). Big data and a lot of computing power are needed to make data-driven systems that work.

Existing methods have various weaknesses and our proposal

The existing approaches to predictive maintenance, which are model-based and data-driven, have a number of limitations; hence, they will present an overview of the most significant shortcomings. They suggest the development of solutions for predictive maintenance that make use of hybrid approaches and are founded on three primary and important aspects in order to effectively address these concerns. They also give a potential use-case scenario for consideration.

IV LIMITATIONS OF CURRENT APPROACHES

A method that is driven by data produces results that are of higher quality. Model-based approaches provide an easy interpretation because the parameters that respond to physical occurrences within systems coincide with behavioral models. This makes it possible for these approaches to provide clarity. Nevertheless, the generation of correct models sometimes proves to be challenging in practice, particularly when dealing with complex systems that are characterized by the presence of a wide variety of physical events. In most cases, even when such a model does exist, it is typically a representation of specific physical phenomena that were produced under precise experimental conditions. As a result, conducting trials under a variety of operating conditions can be a costly endeavor, which restricts the possible application of this approach. On the other hand, the broad availability of sensors and the increased processing power have made it easier to implement artificial intelligence approaches, which has resulted in the development of data-driven methodologies. These methods use artificial intelligence-supported tools to transform monitoring data into behavioral models. Data-driven approaches are a middle ground between being too hard to adopt and being too expensive, accurate, and useful. Methods that are based on data are better than methods that are based on models for systems where it is possible to collect monitoring data that correctly shows how degradation happens. On the other hand, one problem with data-driven methods is that learning can take a long time. While model-based methods yield more precise results, data-based methods are simpler and therefore more practically applicable.

V CASE STUDY

Neurosymbolic AI is gaining traction in various industries, including software development, where it offers unique advantages in combining the strengths of neural networks and symbolic reasoning. Here are a few case studies highlighting the application of neurosymbolic AI in the software industry:

Code Generation and Bug Detection:

Software developers can use neurosymbolic AI to enhance code generation and bug detection processes. By integrating neural networks for learning code patterns and symbolic reasoning for semantic analysis, developers can create more robust code generation tools. For example, a study conducted by researchers at Microsoft Research and the University of California, Berkeley, utilized neurosymbolic techniques to automatically generate code snippets from natural language descriptions. The system employed neural networks to understand the natural language input and symbolic reasoning to translate it into executable code, improving the efficiency and accuracy of code generation tasks.

Automated Software Testing:

We can also apply neurosymbolic AI to automate software testing processes, thereby enhancing the effectiveness of quality assurance efforts. By combining neural network-based techniques for identifying potential software bugs with symbolic reasoning for generating test cases and verifying program correctness, developers can streamline the testing process and improve software reliability. Researchers at Google and Stanford University, for instance, used neurosymbolic techniques in a case study to automatically generate test cases for software programs. The system utilized neural networks to identify potential bugs in the code and symbolic reasoning to generate test inputs that trigger those bugs, enabling more comprehensive software testing.

Natural Language Processing in Software Development:

Neurosymbolic AI has applications in natural language processing (NLP) tasks relevant to software development, such as code summarization, documentation generation, and requirement analysis. By combining neural network-based models for language understanding with symbolic reasoning for semantic interpretation, developers can improve the efficiency and accuracy of NLP-based software development tools. For example, a study conducted by researchers at Facebook AI and the University of Oxford utilized neurosymbolic techniques to automatically generate code documentation from natural language descriptions. The system employed neural networks to extract relevant information from the natural language input and symbolic reasoning to structure the documentation in a human-readable format, facilitating better code understanding and maintenance.

VI CONCLUSION

Neurosymbolic AI presents a promising approach for addressing complex challenges in the software industry, offering a unique combination of neural network-based learning and symbolic reasoning techniques. By integrating these methodologies, neurosymbolic AI enables more effective code generation, bug detection, automated testing, and natural language processing in software development. Neurosymbolic AI bridges the gap between data-driven and rule-based approaches by leveraging neural networks for pattern recognition and learning from large-scale data, as well as symbolic reasoning for logical inference and knowledge representation, resulting in more robust and interpretable software development tools and processes. The case studies highlighted in this discussion demonstrate the diverse applications of neurosymbolic AI in software development, from improving code generation efficiency and bug detection accuracy to automating software testing and enhancing natural language processing tasks. The neuro-symbolic techniques in the subject of predictive maintenance have demonstrated a great deal of promise. The goal is to do additional research and development on these methods, which have the potential to overcome some of the constraints that are associated with conventional model-based and data-driven approaches. The ultimate objective is to successfully supply revolutionary neuro-symbolic models for predictive maintenance in order to make use of new architectures that prioritize interpretability and resilience while also maintaining high levels of performance.

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, Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalch- brenner, 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 re- inforcement 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: dis- entangling 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.