Journal of Reliable Intelligent Environments (2024) 10:257–279
https://doi.org/10.1007/s40860-024-00231-1

REVIEW

Surveying neuro-symbolic approaches for reliable artificial intelligence of things

Zhen Lu[1] · Imran Afridi[2] · Hong Jin Kang[3] · Ivan Ruchkin[4] · Xi Zheng[1,2]

Received: 24 May 2024 / Accepted: 8 July 2024 / Published online: 26 July 2024

Abstract
The integration of Artificial Intelligence (AI) with the Internet of Things (IoT), known as the Artificial Intelligence of Things (AIoT), enhances the devices' processing and analysis capabilities and disrupts such sectors as healthcare, industry, and oil. However, AIoT's complexity and scale are challenging for traditional machine learning (ML). Deep learning offers a solution but has limited testability, verifiability, and interpretability. In turn, the *neuro-symbolic paradigm* addresses these challenges by combining the robustness of symbolic AI with the flexibility of DL, enabling AI systems to reason, make decisions, and generalize knowledge from large datasets better. This paper reviews state-of-the-art DL models for IoT, identifies their limitations, and explores how neuro-symbolic methods can overcome them. It also discusses key challenges and research opportunities in enhancing AIoT reliability with neuro-symbolic approaches, including hard-coded symbolic AI, multimodal sensor data, biased interpretability, trading-off interpretability, and performance, complexity in integrating neural networks and symbolic AI, and ethical and societal challenges.

Keywords AIoT · Neuro-symbolic · Interpretability · Testability · Verifiability

1 Introduction

In the present era of technology, the maturation of artificial intelligence (AI) has reached a pivotal point, where its integration with the IoT is not just a theoretical possibility but a burgeoning reality. This fusion has spawned a cutting-edge area known as *Artificial Intelligence of Things* (AIoT) [1]. AIoT expands both communication and internet technologies, harnessing sensing and intelligent devices to gain a deeper understanding of the physical world. By leveraging these technologies, AIoT facilitates a comprehensive network where devices are not only interconnected but also capable of intelligently analyzing and acting on vast streams of data. This interconnectivity bridges the gaps in human-object and object-object interactions, fostering a continuous exchange of information. The overarching objective of AIoT is to revolutionize how we interact with the physical environment. By enabling real-time control and precise management, AIoT empowers us with informed decision-making, such as commuter behavior prediction in real-time [2]. This synergy between AI and IoT promises a future where the physical world is intelligently orchestrated and optimized for efficiency and innovation.

According to McKinsey [3], IoT technology will have an annual economic impact of several *trillion* US dollars by 2025. This enormous footprint is spread throughout several industries, with the healthcare sector accounting for the highest portion (41%), followed by the industrial and oil sec-

Zhen Lu and Imran Afridi contributed equally to this work.

✉ Xi Zheng
james.zheng@mq.edu.au

Zhen Lu
D23092110348@cityu.edu.mo

Imran Afridi
imran.afridi@hdr.mq.edu.au

Hong Jin Kang
hjkang@cs.ucla.edu

Ivan Ruchkin
iruchkin@ece.ufl.edu

[1] Faculty of Data Science, City University of Macau, Estrada de Coelho do Amaral, Macau 999078, China

[2] School of Computing, Macquarie University, Balaclava Rd, Sydney, NSW 2109, Australia

[3] Computer Science, University of California Los Angeles, 404 Westwood Plaza, Los Angeles, CA 90095, USA

[4] Electrical and Computer Engineering, University of Florida, 1889 Museum Rd, Gainesville, FL 32611, USA

tors (33% and 7%, respectively). Other industries, including retail, public infrastructure, security, irrigation, and transportation, account for roughly 15% of the IoT market. These forecasts highlight the significant and rapid growth of IoT services, together with rising data generation and related economic demands in the upcoming years. That study also highlighted the significance of automated data processing with ML.

As IoT takes hold, the number of connected heterogeneous devices is surging, including UAV systems [4], smartphones [5], autonomous cars [6], smart meters [7], and many more generating thousands of gigabytes per second. Due to the vast amount, complexity, and variability of that data, extracting meaningful insights and making efficient decisions has become increasingly challenging. Initially, these issues were tackled with traditional ML algorithms such as K-means, Decision Trees, Support Vector Machines, and Random Forests. These algorithms demonstrated significant performance in many classification, regression, and clustering problems, including human activity recognition [8], traffic congestion forecasting [9], IoT environment anomaly detection [10], heart disease prediction [11], and general healthcare system [12].

Traditional ML approaches often apply feature engineering and dimensionality reduction methods to optimize performance and resource utilization. Despite their efficacy across various domains, conventional ML approaches encounter adaptability and scalability barriers when applied to more complex IoT applications. Focused on well-defined or hand-crafted features, these systems are dependent on prior domain knowledge. Typically, machine ML components used in these systems are based on the so-called "shallow architectures", which cannot model and represent particularly high-dimensional or complex data. The rich raw data from diverse IoT applications was therefore in dire need of more powerful analytical tools.

The limitations of traditional ML have spurred research into deep learning (DL). For IoT applications, it is an appealing substitute due to its capacity to construct hierarchical representations and automate feature learning. These capabilities enable intelligent decision-making in IoT systems via more reliable and accurate analytics, minimizing human effort. DL performs remarkably well in handling the nonlinear relationships present in IoT data, enabling deeper comprehension by capturing minute variations within intricate multidimensional systems. Consequently, DL models prove to be a superior option for resolving a wide range of essential functions in IoT systems such as heart disease diagnosis [13] and IoT network intrusion detection [14].

Although DL has greatly advanced AI applications, it faces significant issues with *testability*, *verifiability*, and *interpretability*. These problems hamper further applications, particularly critical ones like autonomous systems, smart manufacturing, and finance. To motivate further discussion, we highlight several studies about the issues of testability, interpretability, and verifiability of DL models. In particular, Huang et al. [15] explored certification (i.e., testing and verification) and explanation (i.e., enhancing interpretability) for reliable DL. They emphasized the need for certification before deployment for proper operation, as well as the process of explaining unusual behavior. Similarly, Zhang et al. [16] analyzed testing approaches for ML systems, highlighting the importance of testing in maintaining their trustworthiness. On a related note, Xiang et al. [17] examined the recent developments in integrating DL into safety-critical cyber-physical systems, emphasizing verification, validation, and formal methods to ensure the security of neural network-integrated systems. Chakraborty et al. [18] discussed how trust is based on the interpretability of machine decisions, requiring insights into the internal working mechanisms of these systems. Finally, Zhang et al. [19] discussed the need to improve the interpretability of neural networks in safety-critical situations, specifically for NN-based control software to be accurate, comprehensive, error-free, and fault-tolerant. This study concluded precise testing settings and robustness against frequent failures require further investigation. The existing contributions to testability, verifiability, and interoperability, generally limited because the fundamental issues lie *within the DL models themselves*.

Our study aims to thoroughly review the state-of-the-art (SOTA) DL models and their limitations, particularly in terms of testability, verifiability, and interpretability, the lack of which jeopardizes the reliability of AIoT systems. Next, we examine the emerging *neuro-symbolic paradigm*, focusing on its potential to mitigate these limitations. Neuro-symbolic approaches provide a hybrid framework that combines the robustness and clarity of symbolic AI with the adaptability and performance of DL. These techniques enable AI systems to reason, make justified decisions, and generalize knowledge from large datasets effectively. By integrating neural networks and symbolic AI, neuro-symbolic algorithms offer a promising solution for creating more robust and interpretable AI systems. However, we also identify the remaining *neuro-symbolic challenges*: hard-coded symbolic AI, lack of support for multimodal sensor data, complexity of neuro-symbolic integration, biased interpretability, trade-offs between interpretability and performance, and ethical and societal issues.

Specifically, our survey investigates three research questions:

- *RQ1. What are the primary factors contributing to testability challenges, verifiability issues, and the absence of interpretability in DL algorithms?*
- *RQ2. How do neuro-symbolic algorithms address the challenges of DL models in the AIoT context?*

- *RQ3. What are the key challenges and research opportunities in enhancing AIoT reliability with neuro-symbolic approaches?*

To better understand this promising direction, this paper reviews neuro-symbolic approaches and their applications, focusing on their potential to address the challenges faced by DL models in AIoT. The main contributions of our paper are:

- Identification of SOTA deep learning models used in AIoT applications and an in-depth exploration of their limitations.
- Examination of how neuro-symbolic methods can offer solutions to the issues in DL models.
- Identification of key challenges and research opportunities in enhancing AIoT reliability with neuro-symbolic approaches.

The rest of the paper is organized as follows. Section 2 overviews SOTA DL models used in IoT, the challenges of which are outlined in Sect. 3. Section 4 introduces the neuro-symbolic paradigm, including the evolutionary stages of neural and symbolic AI, SOTA approaches, applications, and their key advantages. Finally, Sect. 5 describes the remaining challenges of neuro-symbolic techniques in AIoT.

2 State-of-the-art deep learning models in IoT

The capability of DL techniques to handle high-dimensional data has drawn a lot of interest in recent years. Multiple-layer neural networks are employed in these techniques to identify and extract significant patterns and characteristics from the data. It has completely transformed the idea of ML, advanced AI, and human–computer interaction to an unprecedented level. IoT applications may improve their capabilities by leveraging DL approaches in several sectors, including smart cities, smart transportation, UAVs, cyber security, smart industries, healthcare, and more. The most prominent DL architectures are discussed below.

2.1 Convolutional neural networks (CNNs)

In DL, CNN algorithm has become a cornerstone for many applications including IoT. It consists of input, output, and hidden layers. The hidden layer includes convolutional, pooling, and fully connected layers. Each layer has a specific function in extracting and learning important features from input data. Convolutional layers convolve input images using a collection of learnable filters or kernels to create feature maps that represent local structures and patterns. After that, pooling layers down-sample these feature maps, keeping key features while shrinking their spatial dimensions. This improves computing efficiency and makes them more resilient to translation variance. The architectures of CNNs have evolved significantly over the years, resulting in enhanced performance in a number of applications, including IoT. IoT applications benefit greatly from MobileNets [20], which are lightweight CNN architectures made specifically for mobile and embedded devices. In order to minimize computational complexity and model size, they use depth-wise separable convolutions, which makes it possible for IoT devices with limited resources to perform effective object identification and image recognition. Another CNN architecture that achieves accuracy equivalent to larger models with a substantially smaller number of parameters is called SqueezeNet, which was proposed by Iandola et al. [21]. It is appropriate for IoT devices with limited memory and processing resources. With its effective "Inception modules", the Inception/GoogLeNet architecture-first proposed by Szegedy et al. [22] enables extensive feature extraction at a lower computational cost, enabling real-time object detection and image classification in IoT devices.

CNNs, therefore, constitute an effective tool for the analysis and interpretation of visual data in IoT systems. It offers an enormous number of intelligent applications in diverse fields, ranging from industrial automation and healthcare diagnostics to smart surveillance and autonomous vehicles. Table 1 showcases some exemplary IoT applications where CNNs have been deployed.

2.1.1 Recurrent neural networks: RNNs and LSTM

Recurrent neural networks (RNNs) is designed to handle sequential data, such as speech, text, or time series data. The basic idea of RNNs is to have an internal state that stores information from prior inputs, enabling the network to learn and model relationships between successive components. RNN designs have been the subject of several variations to solve various issues and enhance performance. The long short-term memory (LSTM) architecture developed by Hochreiter et al. [33] is a well-known RNN variation that solves the vanishing gradient issue that conventional RNNs have. By controlling the input flow, gating mechanisms and memory cells allow LSTMs to efficiently learn long-term dependencies. Another variation, called gated recurrent units (GRUs), proposed by Cho et al. [34] simplifies and increases the computational efficiency of the LSTM design by integrating the input and forget gates into a single update gate. Bidirectional RNNs (BRNNs) [35] have the capability to extract context from inputs received in both the past and future directions due to their ability to analyze sequences in forward and backward directions. Furthermore, by decoupling the neurons inside each layer, the indepen-

Table 1 CNNs in IoT's applications

Category	References	Applications
Biomedical data analysis	[23]	Skin cancer
Smart home	[24]	Home security
	[25]	Human activity recognition
Smart Cities	[26]	Enhanced security
	[27]	Smart energy management
Autonomous vehicles	[28]	Object detection
	[29]	Security
Agriculture	[30]	Hydroponics
Sports	[31]	Motion recognition in sports
Health inspection	[32]	Falling detection

Table 2 RNNs and LSTMs in IoT's applications

Category	References	Model	Applications
Air quality and temperature	[40]	RNN	Temperature and air quality prediction
Smart home	[41]	LSTM + RNN	Traffic flow prediction
	[42]	LSTM	Activity monitoring
Smart city	[43]	RNN	Real-time parking prediction
Intelligent transportation	[44]	LSTM	Intrusion detection
CyberSecurity	[45]	LSTM	Botnet attacks detection

Table 3 RBMs in IoT's applications

Category	References	Applications
Biomedical data analysis	[48]	Skin lesion detection
Intelligent transportation	[49]	Traffic congestion prediction
Cyber security	[50]	Intrusion detection

dently recurrent neural network (IndRNN) [36] solves the gradient vanishing and exploding issues and enables more resilient training with non-saturated activation functions.

In IoT applications, multiple architectures of RNNs, particularly LSTM networks, are used to capture temporal relationships and model sequential data. Time-series forecasting [37], anomaly detection [38], and natural language processing [39] are all tasks that RNNs and LSTMs excel at in IoT environments due to time-varying patterns and correlations in data streams. Numerous IoT applications utilize RNNs and LSTMs; several examples are listed in Table 2.

2.1.2 Restricted Boltzmann machines (RBMs)

Restricted Boltzmann machines (RBMs) [46] are a class of stochastic neural networks designed for unsupervised learning. They consist of visible and hidden layers that are fully interconnected, yet they lack intra-layer connections, which is why they are termed "Restricted". This category of algorithms has demonstrated effectiveness in extracting meaningful patterns from sparse and noisy data typical in IoT environments. For instance, a type of RBM proposed in [47] may be trained in a greedy layer-wise fashion and utilized as building blocks for deep neural networks (DNNs) and deep belief networks (DBNs). With this approach, deep models may be efficiently trained on data of various resource-constrained IoT devices. Some of the IoT applications where RBMs have shown their significance are mentioned in Table 3.

2.1.3 Autoencoders (AEs)

An Autoencoder is a type of neural network that is specifically designed for unsupervised learning, with applications ranging from feature learning to data compression and denoising. An Autoencoder is comprised of encoder, bottleneck, and decoder. The main principle of Autoencoders is to take in input data, compress it, and then use that compressed representation, or encoding, to recreate the original data. The input data is compressed by the encoder into a representation in a lower dimension (bottleneck) and the compressed representation is used by the decoder to recreate the original data.

Table 4 Autoencoder in IoT's applications

Category	References	Model	Applications
Wireless sensor network	[55]	AE	Anomaly detection
	[56]	Spatio-temporal AE	Signal reconstruction
Smart home	[57]	AE	Energy consumption
Intelligent transportation	[58]	Denoising-AE	Data sampling
	[59]	AE + LSTM	Anomoulous event detection
	[60]	AE	Traffic congestion prediction
Smart manufacturing	[61]	AE	Intrusion detection in smart factory

Table 5 GNN in IoT's applications

Category	References	Applications
Autonomous driving	[65]	Trajectory prediction
Energy management	[66]	Performance prediction of power station
Object detection and tracking	[67]	Object tracking
Robotics	[68]	Decentralization of control
Remote sensing	[69]	Aerial image classification
	[70]	Object classification
Smart transportation	[71]	Travel time estimation
Human activity detection	[72]	Activity recognition
Neural network	[73]	Energy prediction

The goal of the Autoencoder's training process is to reduce the reconstruction error that is, the discrepancy between the input and reconstructed data. One type of AEs that works well for feature extraction and dimensionality reduction in IoT sensor data is the sparse autoencoder [51], which imposes sparsity restrictions on the hidden layer activations during training. Denoising autoencoders [52] are effective for data denoising in IoT applications because they can be trained to recover the original input from degraded data. This makes them noise-resistant. In order to efficiently compress and generate new data samples, variational autoencoders (VAEs) [53] learn the underlying probability distribution of the input data. This is advantageous for data transmission and augmentation in IoT networks. Convolutional autoencoders [54], which use convolutional layers, are useful for applications like predictive maintenance and video anomaly detection because they can efficiently extract temporal and spatial correlations in IoT sensor data. These autoencoder variations have made it easier to construct reliable and effective ML models for a range of IoT applications, addressing issues such as data scarcity, privacy concerns, and resource constraints. Several applications of Autoencoders in IoT are listed in Table 4.

2.1.4 Graph neural networks

Graph-structured data is common in many domains including social networks, biological systems, recommendation systems, and IoT networks. Such data is handled by graph neural networks (GNNs), a family of DL models. By extending typical neural network architectures to take graphs as inputs rather than grid data, GNNs can track complicated associations among entities and events. The GNNs consist of nodes, edges, and message-passing mechanisms. Every node in a graph is initially associated with a feature vector describing its attributes. The nodes then aggregate information from each other in each iteration through a process called message passing, where information is passed from neighboring nodes. Nodes aggregate messages received from neighbors and combine them with their features to update their representations. To update this representation, a neural network layer is applied or a pooling operation is performed. During this process, nodes continuously refine their representations based on information they receive from their neighbors over a fixed number of iterations or until convergence is achieved. The final representation of the nodes represents different tasks such as classification and prediction. GNN has variant flavors, such as graph convolutional network (GCN) [62], variational graph autoencoder [63], and variational graph recurrent neural network [64]. Various applications of IoT have leveraged the GNN, some of the particularly advantageous ones are listed in Table 5.

2.1.5 Geometric deep learning models

Geometric neural network models can process and analyze geometric or graph-structured data efficiently [74]. These

Table 6 Geometric deep learning in IoT's applications

Category	References	Applications
Smart industries	[83]	Improving industry's lean management system
Energy management	[84]	Power load forecasting
Molecular modeling	[85]	Drug discovery
Human computer interaction	[86]	Algorithm optimization
Smart transportation	[87]	Traffic speed prediction
Neural network	[88]	Optimization of CNNs
Video processing	[89]	Emotion recognition

models primarily use manifold and graph data. The network structure data's nodes and edges make up the graph. For example, in a social network, each node represents a person's information and each edge represents a link between people. The manifold data make it clear to understand the high dimensional data where a lot of points are scattered in a 3D space.

Data can be randomly spread out, which makes it challenging for an algorithm to discover hidden patterns. Geometric Deep Neural Network models can recognize and make use of the inherent structure and patterns found in the data such as cloud [75], meshes [76] and manifolds [77]. Since they are built to consider the spatial relationships and connectivity between data points. These models adopt a mechanism that can run on irregular and geometrically structured data by utilizing concepts of computational topology [78], differential geometry [79] and graph theory [80]. Developing methods for efficiently aggregating and propagating information across irregular data structures is a major difficulty in geometric DL.

Several geometric DL models operating on symmetric positive definite (SPD) manifolds have been proposed, such as Tensor-CSPnet [81], which is designed for classifying MI-EEG data. Tensor-CSPnet utilizes special linear mapping and ReLU layers that operate on SPD manifolds, directly mapping the SPD matrix points to the tangent space for Euclidean distance operations, such as convolution and fully connected operations. This manifold learning approach enables direct processing of the covariance matrix of the input sensor data, thus preserving the original data attributes and features. Applications of geometric DL models can be found in many fields. For instance, GCNNmatch [82] is a geometric DL model for multi objects tracking (MOT). The initial phase in GCNNMatch is to extract each identified object's appearance feature map h_{pp}. This is accomplished by inserting the corresponding bounding box into a CNN. The FC layer is omitted, resulting in a high dimensional temporary output feature map. The definition of a geometric feature of an object is $h_{geom}=(a,b,c,d)$, where (a,b) represents the bounding box's position coordinates, c for width, and d represents height. In the next stage, the object's edges e^z features are extracted by merging h_{pp} and $h_{geom}=(a,b,c,d)$ and inserting the merged vector into the FC layer f_{edge}. In order to include contextual information from nearby nodes during the feature extraction process, the node and edge features $h^v = h_{pp}$ and e^z are finally fed into a GCNN with two hidden layers. Some most recent applications of Geometric DL in IoT are detailed in Table 6.

2.1.6 Transformer deep learning models

Introduced by Vaswani et al. [90], the Transformer model entirely replaced traditional recurrent and convolutional neural networks with a novel architecture that relies only on self-attention mechanisms, revolutionizing the field of natural language processing (NLP). Since then, many advanced NLP models [91] have included this architecture as a benchmark model. The model consists of a decoder and an encoder. The encoder takes in a sequence of tokens (for example, words or characters) and outputs a continuous representation of the input sequence. From the encoder's output and previous tokens, the decoder generates the output sequence one token at a time. Both encoder and decoder consist of identical layers, each sub-divided into two sub-layers: a self-attention mechanism, which computes attention weights that represent the relative importance of each token in the input sequence concerning every other token, followed by a feed-forward network (FFN), which transform the output of the self-attention mechanism into a higher-dimensional space. The encoding, which consists of sine and cosine functions of various frequencies, is applied to the input embeddings. Similarly, by using various attention mechanisms at once, the Transformer model's multi-head attention enables the network to capture a variety of relationships and dependencies among the input sequence. By concentrating on distinct input segments, each attention head helps the model acquire more robust and expressive representations. This method improves generalization to a variety of input patterns, lessens information bottlenecks, and increases the model's capacity to effectively handle a broad range of natural language processing tasks. There are many variants of transformer including the most prominent one is BERT (bidirectional encoder representations from transformers) [92], which was

Table 7 Transformer models in IoT's applications

Category	References	Applications
Smart home	[94]	Spoken notification generation
IoT data streams	[95]	Device identification
Wireless communication	[96]	Automatic modulation recognition
Smart grid	[97]	Anomaly detection
Security	[98]	Intrusion detection in MQTT protocols

pre-trained on a huge corpus of textual data. On a variety of natural language processing (NLP) tasks, such as named entity recognition, text categorization, and question answering, BERT and its variants such as RoBERTa [93] have demonstrated state-of-the-art performance.

Some of the areas where transformer models are used in IoT applications are mentioned in Table 7.

After investigating several DL architectures and their uses, it is clear that DL has advanced significantly in resolving challenging problems in a variety of fields. To reach its full potential, DL must overcome several obstacles and constraints that come with these developments. We explore some of the major concerns surrounding DL in the section that follows. These challenges include testability, interpretability, and verifiability. To create DL systems that are more dependable and trustworthy, these issues must be recognized and addressed.

3 RQ1: challenges of deep learning architectures

Advances in DL have transformed many research fields, often with performance surpassing that of traditional approaches. The complexity and widespread adoption of DL models, however, have led to concerns about their testability, verifiability, and interpretability. For example, autonomous driving [6] and health care [23], where these models can have major implications for human lives, are among the most high-stakes domains where these challenges have become increasingly important. A comprehensive approach to addressing these issues will ensure that DL technologies can be widely adopted and developed to their full potential. Zhang et al. [16] carried out an in-depth analysis of ML testing approaches, including definitions, distributions of research, datasets, and market trends. It emphasizes the need for rigorous testing procedures to guarantee reliable performance and the crucial role that ML testing plays in maintaining the credibility of ML systems.

3.1 Testability

As DL models gain traction across multiple domains, *testability* becomes an increasingly important challenge. Within the field of ML, testability pertains to the capacity to conduct systematic testing to fully assess the safety, robustness, and accuracy of a model's behavior. This is crucial for DL in particular, which has shown impressive results in challenging tasks but frequently acts as a "black box" with incomprehensible decision-making procedures. DL models pose distinct issues because of their intrinsic complexity, sensitivity to data quality, and absence of standardized testing frameworks, in contrast to typical software systems where testability is a well-established practice. To increase trust and achieve the complete potential of DL, these testability issues must be addressed. In the following section, we will explore the core issues that hinder the testability of DL algorithms.

Complexity and scalability DL algorithms are complicated because of their huge parameter sets, sophisticated architectures, and computationally demanding training procedures. Neural networks are made up of several layers of connected nodes that individually process intricate calculations on input data. These models contain up to millions or even billions of parameters these models frequently contain. Since DL models frequently function as closed systems, it might be difficult to comprehend how they make decisions internally [99]. The inability to fully test and validate the behaviors of the models is caused by this opacity.

DL algorithms are highly scalable [100]. Scalability, in this context, refers to the model's capacity to effectively manage massive amounts of data, intricate topologies, and computing resources. One key aspect of scalability is the ability to efficiently handle enormous datasets, including millions or even billions of samples, by leveraging distributed or parallel computing paradigms. However, the scalability of testing efforts may be hindered by the prohibitively high computational costs required for comprehensive testing of such large-scale DL models [101]. This issue poses a significant challenge in ensuring the robustness and reliability of DL models across various applications and domains. To address these challenges, ongoing efforts are directed towards the development of efficient model architectures, algorithm optimizations, and the utilization of distributed computing platforms [102].

Reliance on data quality and quantity

Since DL models are data-driven, the quality and accuracy of the training dataset significantly impact their performance

[103]. To ensure the validity of these models, the training datasets used should be testable, as they are susceptible to biases and anomalies [104]. One of the crucial aspects of IoT is the data quality, especially when it comes to mission-critical AIoT systems such as autonomous cars. The reliability and accuracy of data are essential for ensuring the effective functioning of such systems [105].

Coverage challenges An increasingly standard approach is needed to assess and test DL models as their significance grows with time. New methods, such as coverage testing and metamorphic testing, are being investigated because it is possible that traditional testing methodologies may not be sufficient. These testing methods despite their potential, still encounter significant challenges. Guiding test generation using coverage does not always succeed in improving test suite effectiveness and generates more biased predictions [106]. On the other hand, metamorphic testing is a manual process in which the users have to specify metamorphic relation and does not guide the exploration of a model's behavior [107].

Specification challenge To ensure the reliability as well as safety of DL-based AIoT in real-world scenarios, extensive testing is necessary. To detect possible vulnerabilities and guarantee system resilience, thorough testing procedures are crucial due to the innate complexity and non-deterministic behavior of DL models. This means creating extensive test suites that cover a wide range of scenarios and edge cases, such as changes to input data, system states, and environmental factors [108]. A testing framework is proposed in [109] for autonomous driving in uncertain environments, leveraging deep reinforcement learning for falsifying STL (Signal Temporal Logic) formulas and demonstrating efficacy through case studies in the autonomous driving domain. STL is a formal specification language used to describe the properties and requirements associated with real-time signals that change over time. However, one unsolved challenge is the manual extraction of STL formulas, which is error-prone and laborious [103].

3.1.1 Interpretability

Interpretability refers to the user's ability to understand and explain how a model makes decisions in a way that is meaningful and accessible [110]. The inherent complexity and opacity of DL systems make them hard to interpret. The stakeholders (domain experts, policymakers, and the public) should have a clear understanding of how these models make their predictions to create trust, ensure accountability, and make responsible deployment possible. Therefore, resolving interpretability issues is essential to utilize the full potential of these effective approaches and guarantee their reliable usage. Below, we describe some of the most significant issues that prevent interpretability.

Lack of standard evaluation metrics The lack of benchmark assessment techniques to assess interpretability approaches is a major obstacle to attaining interpretability for DL models. Because the exact definitions of the terms interpretability, explainability, transparency, and other related concepts are still up for discussion and disagreement [110]. Because there are no standard procedures, it is difficult to develop objective criteria for assessing and comparing the interpretability of various DL architectures.

Accuracy vs interpretability trade-off The relationship between interpretability and model accuracy is a complicated and widely discussed issue. On one side of the debate, it's frequently stated that if models become less interpretable, they become more accurate and sophisticated because they learn complex patterns in the data [111]. This trade-off is especially noticeable when it comes to deep neural networks, which have opaque decision-making processes, and have state-of-the-art performance on many tasks. However, recent studies have challenged the idea that accuracy and interpretability are directly correlated. Research has indicated that it is feasible to create models that are highly interpretable and accurate at the same time. This can be achieved, for instance, by employing strategies like regularization and feature selection or by incorporating prior domain knowledge [112]. There is still much to investigate regarding the accuracy-interpretability trade-off [113] and there is no conclusive agreement on whether this relationship holds for all situations.

Complex decision-making mechanisms Since deep neural networks are the foundation of many cutting-edge DL models, they are frequently referred to as "black-box" systems since it is difficult to determine the exact cause of the model's predictions due to its intricate, multi-layered structure. DL models use enormous volumes of data to learn complex, non-linear relationships, which are difficult to interpret or intuitive to human observers. Deep neural networks' internal workings are opaque, which makes it difficult for stakeholders-including domain experts and end users to grasp how the model generates its outputs. This is a major obstacle to interpretability [18]. Many studies have attempted to "open the black box" and offer explanations for DL predictions. For instance, through the visualization of patterns within the input data that elicit specific responses in the model, researchers can glean insights into the acquired knowledge of the model [114]. Additionally, employing Layer-wise Relevance Propagation (LRP) [115] helps attribute the model's predictions to specific input features. This technique works by propagating relevance scores backward through the network. It enables the researchers to discern the most influential aspects of the input data guiding the model's predictions. Nonetheless, the complexity of these models continues to pose a formidable challenge.

3.1.2 Verifiability

Verifiability is a major issue in the field of DL [15]. Deep neural networks are often criticized for their opaque internal decision-making, which makes them difficult to understand or verify. Because of this opacity, it may be difficult to verify that a DL model's decisions are in line with the desired goals or values [116]. Researchers have put forth strategies to deal with this issue, like creating "verifiable AI" frameworks [117] that include supplementary verification tasks in addition to the main prediction task. The notion is that even if the underlying reasoning is still unclear, the model's decision-making process can be assured to perform well on these verification tasks. Verifiability is a useful addition to explainability in DL systems that can ensure confidence and guarantee desired output. A framework called Alpha-Beta-CROWN [118] is designed to offer rigorous assurances on the robustness of neural networks to the changes in the inputs. The framework uses bounding propagation to generate output using a defined set called "Alpha" and "Beta" sets. The inputs in the Beta set are those for which the network's prediction may vary from the desired output but only within a bounding range. whereas the inputs in the Alpha set are those that are certain to generate accurate predictions. Some of the issues which degrade verifiability are discussed below.

Non-convex optimization The verification of DL networks is significantly hampered by the non-convex optimization problem. DL models include the optimization of highly non-convex objective functions [119], in contrast to convex optimization problems, which have a single global optimum that can be easily discovered. This indicates that it is challenging to ensure convergence to a global optimum due to the abundance of local minima in the loss curves of DL models. Depending on the initialization and optimization procedure employed, the existence of several local minima may result in differences in the behavior and performance of the trained model. Making sure DL models converge to a desired solution that accurately reflects the underlying data distribution is necessary to validate their accuracy. The verification procedure is made more difficult by the non-convex nature of the optimization problem, which makes it even more difficult to tell if the trained model has found a good solution or is trapped in a sub-optimal region of the parameter space. To address the non-convex optimization challenges in DL [120], it is essential to develop reliable optimization techniques, regularization methods, and initialization strategies that minimize the risk of converging to suboptimal solutions, but it is challenging by nature.

Input perturbation limitations In DL, the phenomena wherein minor adjustments or perturbations in the input data result in substantial changes in the model's predictions is known as the sensitivity of input perturbation. DL models frequently exhibit sensitivity to small changes in input data, which may result from adversarial attacks, changes in the input distribution, and noise [121]. The DL model's decision boundary may be impacted by these perturbations, which could lead to inaccurate predictions. To guarantee the durability of DL models in practical applications, it is essential to comprehend and reduce the sensitivity to input perturbations. Methods for measuring how sensitive DL models are to input perturbations have been proposed by researchers. For example, the idea in [122] was created to quantify how changes in the input can affect the output of a CNN. Iterative algorithms can be used to compute this sensitivity, which can be used to evaluate how robust CNNs are to input noise [123]. The capacity of current methods to test the input–output robustness of systems is limited to handle complex system-level properties specified in formal logic such as signal temporal logic (STL). Furthermore, multimodal sensor data inputs-which are frequently encountered in AIoT applications-cannot be effectively verified using these methodologies.

4 RQ2: neuro-symbolic paradigm

As IoT technology continues to advance, AIoT has risen as a pivotal trend shaping the technological landscape of the future. By fusing the prowess of AI with the expansive reach of IoT, AIoT paves the way for sophisticated control mechanisms and insightful data analytics across IoT devices, thereby enhancing convenience and boosting productivity in both personal and professional realms. Neuro-symbolic approaches are gaining attention as a rising force, driving significant advancements and signaling a fresh chapter for AIoT.

4.1 Overview of neuro-symbolic paradigm

The neuro-symbolic paradigm is an approach that combines neural networks (often associated with deep learning) and symbolic artificial intelligence (symbolic AI) in addressing complex cognitive tasks. Neuro-symbolic computational models amalgamate the power of neural networks with the precision of symbolic logic, aiming to harness the strengths of both to enhance the performance of AI systems. These networks endeavor to blend the neural network's proficiency in handling unstructured data, such as text, speech, and images, with the symbolic system's aptitude for managing structured data and engaging in logical reasoning. Currently, with the rapid development of DL, connectionist approaches have achieved significant success in many fields. Simultaneously, neuro-symbolic AI is attempting to harness the advancements in DL to create intelligent systems capable of both DL and symbolic logical reasoning, thus addressing a broader range of complex problems [124]. This integrated approach may

Table 8 Evolutionary stages of neuro-symbolic paradigm

Time	Stages	References	Methods
1990–2000s	Initial concepts	[125, 126]	KBANN, CLIP
2010s	Development	[127–129]	Conceptors, NTMs, memory networks
2010–2020s	Current trends	[124, 130]	Multimodal reasoning models

Table 9 Categories of state-of-the-art neuro-symbolic approaches

Category	References	Methods
Knowledge graph representation learning	[131]	KGEs
Semantic parsing	[132]	NSP
Logical reasoning	[133]	DeepProbLog
Program synthesis	[134]	R3NN
Intelligent agents and planning	[135]	Logical neural network
VQA	[124, 136]	NS-CL, N2NMNs
	[130, 137]	NS-DR, ViperGPT

represent a significant direction for the future development of AI. The neuro-symbolic paradigm has undergone three key stages of development: initial concepts, development, and current trends, as illustrated in Table 8. These stages reflect the evolution of research and practical implementations aimed at leveraging the strengths of both approaches.

Initial concepts (1990–2000s). The concept of neuro-symbolic AI can be traced back to the 1980s when many AI scholars began to explore how to combine symbolic logic with the connection mechanisms of neural networks. In 1990, Towell et al. [125] proposed the Knowledge-Based Artificial Neural Network (KBANN), the first system to allow background knowledge in learning within neural networks and knowledge extraction. Garcez et al. [126] introduced the contrastive language-image pre-training system (CLIP) in 1999, where they transformed background knowledge into propositional logic, based on which they constructed a forward artificial neural network, and induced new knowledge from examples to update existing knowledge. However, these methods did not make significant progress due to the limitations of ML technology at the time.

Development (2010s). Entering the 21st century, with the rise of DL, neuro-symbolic AI has been presented with new development opportunities. Researchers began to attempt to combine DL models with symbolic logic reasoning to enhance the models' interpretability and reasoning capabilities. In 2014, Jaeger proposed a method of controlling recurrent neural networks with "Conceptors" [127], which endowed the entire network with geometric properties and enabled effective integration with Boolean logic. Graves et al. introduced neural turing machines (NTMs) [128] in 2014, and Sukhbaatar et al. proposed memory networks [129] in 2015, both of which incorporated memory mechanisms to address the issue of storing intermediate results in the reasoning process.

Current Trends (2010s–2020s). In recent years, the field of neuro-symbolic AI has made remarkable progress, particularly in knowledge graph representation learning [131], semantic parsing [132], logical reasoning [133], program synthesis [134], intelligent agents and planning [135], and visual question answering (VQA) [124, 130, 136, 137]. Neuro-symbolic AI represents a promising direction in the evolution of AI, aiming to combine the best of neural networks and symbolic reasoning to create powerful and interpretable systems. The field is rapidly evolving, with significant research advances, practical applications, and growing interest from both academia and industry.

Human knowledge describes the entire world experienced by humans. This knowledge can be quickly generalized to different new problems. Neuro-symbolic AI aims to integrate neural networks with human understanding, ultimately mastering a knowledge system akin to a human's, while not losing the flexibility of neural networks.

4.2 Overview of state-of-the-art neuro-symbolic approaches

Neuro-symbolic AI is an active and promising field within the realm of AI [138]. Its development trend is to more deeply integrate DL and symbolic reasoning to overcome their limitations in representation learning and reasoning. The latest research develops more powerful and flexible models that can balance the strengths of DL and symbolic reasoning when dealing with different tasks. Recent neuro-symbolic approaches can be categorized into six main types based on the task at hand: knowledge graph representation learning, semantic parsing, logical reasoning, program synthesis, intelligent agents and planning, and VQA, as shown in Table 9.

Knowledge graph representation learning. Knowledge graphs (KGs) [139] are structured, semantic frameworks

that represent real-world entities (such as people, places, and organizations) and the relationships between them. They organize knowledge in a machine-readable format, which is useful for applications like data integration, information retrieval, and natural language processing. Neuro-symbolic approaches in knowledge graphs integrate symbolic reasoning, which uses logic and defined rules, with sub-symbolic methods like neural networks, which excel at processing large-scale, noisy data. This aims to leverage the strengths of both techniques to enhance knowledge graph reasoning. For example, Wickramarachchi et al. [131] utilized knowledge graph embeddings (KGEs) to manage the vast amount of heterogeneous data generated by vehicle sensors in the field of autonomous driving (AD). The performance of KGEs on autonomous driving data was evaluated, and the research explored the relationship between the level of informational detail in a knowledge graph and the quality of its derivative embeddings. Different KGs with varying levels of informational detail were generated for different datasets. The purpose of these KGs was to represent the various scenarios or situations that an autonomous vehicle encounters on the road. The study demonstrated that more detailed KGs are better at capturing type and relational semantics, while also raising important questions about the applicability of evaluation metrics used in existing literature. Additionally, it opened up a rich field for future research in neuro-symbolic AI within the IoT.

Semantic parsing. Semantic parsing is a subfield of NLP and AI that aims to understand the meaning of natural language sentences in a way that is amenable to computational processing. It involves converting natural language text into a formal, structured representation, often in the form of a semantic graph or a logical form that captures the underlying meaning of the text. Over the past few years, significant research has been into neuro-symbolic AI's applications in natural language processing (NLP) and semantic analysis. A key focus has been on how the integration of symbolic reasoning with neural network-based DL can address some of the limitations in current NLP technologies. These limitations include the lack of robustness in understanding context and performing abstract reasoning, which is crucial for advanced semantic analysis. One of the significant contributions in this field has been examining the suitability of neuro-symbolic AI for various NLP tasks. This includes its role in enhancing model interpretability, leveraging symbolic reasoning for improved data efficiency, and enhancing the understanding of complex language constructs which are often challenging for purely neural models. For example, Liu et al. presented neural-symbolic processor (NSP) [132], a framework for natural language understanding that operates as follows. An encoder converts input text into a text embedding, which predictors then use to produce neural predictions. Simultaneously, decoders transform the embedding into executable programs. The encoder and decoders together form a sequence-to-sequence model. Executors process the programs to create symbolic predictions. Finally, a mixture-of-experts model, guided by a gating network, integrates the neural and symbolic predictions to make the final decision. NSP is powerful in dealing with the challenging problems that conventional neural-network-only approaches suffer from.

Logical reasoning. Logical reasoning is the process of concluding systematic logical rules. It involves using known information and explicit rules to derive new information or conclusions. Neuro-symbolic AI combines the data-driven capabilities of DL with the transparency and structured knowledge representation of symbolic reasoning, enabling more efficient learning and reasoning processes in logical reasoning tasks. For example, Manhaeve et al. [133] contribute to the field of probabilistic logic programming with the introduction of DeepProbLog. DeepProbLog achieves more efficient and accurate reasoning when handling complex data by integrating deep neural networks with ProbLog [140], an established probabilistic logic programming language. In contrast, traditional ML methods typically model the relationship between inputs and outputs rather than reasoning based on logical rules. This implies that they may be unable to perform logical reasoning or handle complex logical structures. The capacity for neuro-symbolic techniques to engage in logical reasoning is, therefore, a crucial factor in developing AI solutions that are not only reliable but also possess a high degree of interpretability.

Program synthesis. Program Synthesis refers to the automated generation or construction of computer programs given specifications or requirements. Traditional neural network architectures, when used for program synthesis [141], often incur high computational costs, are difficult to train, have poor interpretability, or make it challenging to verify their correctness. Previous methods typically required training a separate model for each task (program), which limited the generalizability and scalability of the models. Neuro-symbolic methods can address these limitations of traditional approaches in automatic program generation. For instance, Parisotto et al. [134] proposed a new technology called neuro-symbolic program synthesis (NSPS). The core idea of NSPS is to use two novel neural modules, the cross-correlation I/O network and the recursive-reverse-recursive neural network (R3NN). The cross-correlation I/O network can receive a set of input–output examples and produce a continuous representation of these examples. Given the continuous representation of examples, R3NN synthesizes a complete program by incrementally expanding partial programs. R3NN employs a tree-based neural architecture that sequentially constructs a parse tree by selecting rules from a context-free grammar (i.e., the domain-specific language DSL). For example, to solve string transformation problems based on regular expressions, NSPS can construct programs

from new input–output examples and create new programs for tasks that were never observed during training. Experiments show that the R3NN model can construct programs from new input–output examples and build entirely new programs for tasks never seen during the training process, thus addressing traditional methods' shortcomings in generalization and interpretability.

Intelligent agents and planning. Intelligent agents [142, 143] are computational systems capable of perception, reasoning, and action, enabling them to understand the information in the environment and take appropriate actions to achieve set goals. Planning is a crucial concept in intelligent agents, involving the determination of a sequence of actions or a plan required to achieve an objective. Intelligent agents are closely related to planning as it is a significant method for them to realize their goals. Agents typically use planning to decide on their next course of action and continuously adjust their plans to adapt to changes in the environment. Some methods combine neural networks with symbolic reasoning to address planning and decision-making problems in reinforcement learning (RL), such as model-based reinforcement learning and policy gradient methods. These approaches employ symbolic planning to generate high-level action plans, which are then executed by neural networks to realize the planning and execution capabilities of intelligent agents. For example, Kimura et al. [135] explore a novel RL method for text-based games using a neuro-symbolic framework called logical neural network, which addresses intelligent agents and planning problems by integrating the learning capabilities of neural networks with the reasoning and knowledge representation strengths of first-order logic. This approach enhances learning efficiency, interpretability, and planning abilities, leading to more capable and understandable intelligent systems.

Visual question answering. VQA is a task in the field of AI and computer vision that involves answering questions about images. In this task, a system is given an image along with a natural language question about the image, and it must provide an answer based on the visual content. The questions can range from simple (e.g., "What color is the dog?") to complex (e.g., "What is the man doing in the image?"). VQA systems typically integrate techniques from both computer vision and natural language processing to interpret the visual information in the image and the textual content of the question. This involves using image recognition models to analyze the image, language models to process the question, and a way to combine these two streams of information to generate a correct answer. Below we discuss some of the key works in this area, which is one of the most active in neuro-symbolic techniques.

Mao et al. [136] introduced the neural-symbolic concept learner (NS-CL), which uses a CNN to recognize object attributes and generate a dynamic knowledge base and employs an RNN to convert natural language questions into symbolic form. The NS-CL not only excels in visual question answering and image-text retrieval but also enhances interpretability and learning efficiency under conditions of limited data, showcasing the advantages of Neuro-symbolic methods. Influenced by module networks [144], Hu et al. [124] proposed a model known as end-to-end module networks (N2NMNs). N2NMNs comprise two main components: a set of neural modules with shared attention, providing parametric functions to address subtasks, and a layout strategy that predicts the layout for specific questions and dynamically assembles a neural network from it. N2NMNs employ a tailored set of modules enhanced with a soft attention mechanism, which dynamically supplies textual parameters specific to each module. Dynamic visual reasoning, particularly understanding physical interactions between objects, poses a significant challenge in computer vision. Humans naturally interpret such dynamics using intuitive physics, but equipping AI with similar capabilities, especially in AIoT applications like industrial robotics, is crucial for improving autonomy and operational safety. Based on these, MIT introduced the neural-symbolic dynamic reasoning (NS-DR) model [130], which not only predicts unseen movements but also handles predictive and counterfactual reasoning, establishing a robust framework that integrates vision, language, dynamics, and causality to model complex interactions and reasoning tasks effectively. However, the model's reliance on densely annotated videos for accurate visual and physical attribute representation can pose practical challenges in real-world applications.

The modular systems (N2NMNs, NS-DR) mentioned above attempted to decompose tasks into simpler modules. However, they are difficult to extend to more complex tasks or real-world applications because program generation is highly domain-limited, and training the program generator is difficult. To overcome these bottlenecks, Suris et al. proposed a new framework for visual and language query processing called Visual Inference via Python Execution for Reasoning (ViperGPT) [137]. ViperGPT also adopts a modular approach to visual reasoning, but unlike previous modular systems, it does not require predefined specific functions or training a program generator. Instead, it utilizes the LLMs to generate Python code, composing vision-and-language models into subroutines to produce a result for any query. By defining simple, task-specific APIs, ViperGPT can leverage existing models' code generation and reasoning capabilities without the need for fine-tuning. Additionally, the output of the code generation model is code, which is more interpretable than end-to-end models. The success of LLMs has paved new avenues for the development of neuro-symbolic AI, fostering the integration of symbolic reasoning with neural networks. This has provided novel approaches and

methodologies for addressing complex logical and reasoning challenges.

4.3 Applications of neuro-symbolic approaches in AIoT

The application of neuro-symbolic AI in AIOT is primarily reflected in enhancing the decision-making, reasoning, and interpretive capabilities of intelligent systems by integrating the data processing capabilities of neural networks with the advantages of symbolic logic reasoning. Specific application areas are as follows:

Common-sense autonomous driving. This is a task that requires a common-sense understanding expressed in symbolic terms, which the neuro-symbolic paradigm can greatly help with. The application of neuro-symbolic AI in the field of autonomous driving mainly focuses on enhancing the accuracy and logic of decision-making, as well as improving the interaction between vehicles and the environment. The following works contributed to this area. Doe et al. [145] explored how neuro-symbolic reasoning systems bring benefits and also pointed out the challenges faced when applied in the traffic domain, especially in the context of autonomous driving. The main focus is on how to improve the system's robustness and interpretability by combining neural and symbolic features. JSHST et al. [146] introduced neuro-symbolic program search (NSPS) for the design of decision-making modules in autonomous driving. It is an automated search method for synthesizing neuro-symbolic programs. NSPS is capable of automatically tuning hyperparameters to generate robust and expressive neuro-symbolic programs. Sharifi et al. [147] introduced a model-free deep reinforcement learning (DRL) approach, termed DRL with symbolic logics (DRLSL), which integrates the advantages of DRL-learning from experience-and symbolic first-order logics-knowledge-driven reasoning. This integration aims to facilitate safe learning in real-time interactions for autonomous driving within actual environments. The innovative DRLSL approach actively engages with the physical environment to learn autonomous driving policies, ensuring safety throughout the process.

Service robots with high demands for versatility. Service robots of this kind do not require performance specialized in a single task but rather rely on stable and reliable performance, as well as the ability to be interpretable and communicative. Therefore, understanding the context and mastering human symbols holds extraordinary significance for them. Human concepts are structured as networks composed of symbols, rather than being singular symbols [148]. Of course, we hope that AI can achieve this as well. For instance, when an elderly care robot sees a pear, it needs to understand that it belongs to the biological category of fruit, is edible, and has high water and sugar content. It also needs to know the pear belongs to the category of physical objects, which have attributes such as size, shape, and color. These classes and attributes can be captured by individual neural network classifiers and effectively combined into a graph structure (which can be simply understood as a form of knowledge graph, similar to the structure of a family tree). Once combined, a complete concept of a pear is formed. Constructing these classifiers and binding them with the corresponding graph structures is the foundation for building such a visual concept system. With this system, the pear in front of the robot is not just an object that can be recognized, but a scenario involving a lot of background knowledge that can influence the robot's behavior. It can decide whether to bring the pear to the owner based on the situation. If the pear is sweet and the owner is an elderly person with diabetes, the robot needs to make a decision not to offer it, based on the background knowledge it possesses. There is emerging research in this domain, as discussed below.

Namasivayam et al. [149] introduce a neuro-symbolic learning method to address the task of language-guided robot manipulation. Specifically, the method aims to train a model that can output an executable manipulation program based on natural language instructions and an input scene. Venkatesh et al. [150] proposed a pipelined architecture that integrates object detection, spatial reasoning, and attention mechanisms to address the problem of spatial reasoning and manipulation of objects by robots based on natural language instructions. This architecture consists of two main stages: Initially, in the Localization stage, a separately trained object detector is used to identify and localize all objects within the scene. Following this, in the spatial reasoning stage, the natural language instructions and the localized coordinates of the objects are mapped to the start coordinates from where the robot must pick up the object and the end coordinates where the object should be placed. Hanson et al. [151] introduced a novel anthropomorphic arm controller designed for the Sophia robot. This arm integrates machine perception, convolutional neural networks, and symbolic AI for logical control and affordance indexing.

Systems with highly limited data. AIoT may face the issue of insufficient data samples in certain scenarios, which typically occurs in situations where it is difficult to collect a large amount of data or where the cost of data acquisition is relatively high. Neuro-symbolic AI, due to its characteristics of combining DL with symbolic logic reasoning, can effectively learn and reason under conditions of limited data, enhancing the interpretability of the model, and leveraging transfer learning to address new problems. This capability is of great significance for scenarios with small sample sizes, such as personalized recommendations for specific customer groups [152]. For example, Anup et al. [153] introduced a neuro-symbolic method for predicting the strategies that students employ when solving problems. By predicting stu-

dents' learning strategies, the system can better adapt to the individual learning needs of different students, enhancing the personalization and adaptability of automated instructional systems, such as intelligent tutoring systems, and thereby improving learning outcomes. Fan et al. [154] presented Athena 3.0, a sophisticated multimodal chatbot featuring neuro-symbolic Dialogue Generators, crafted by the University of California, Santa Cruz. Engineered to deliver captivating and coherent dialogue across a spectrum of trending topics, Athena 3.0 serves the Alexa Prize Socialbot Grand Challenge (SGC) with distinction. Beyond its capacity as a voice interaction platform, Athena 3.0 enriches user engagement through multimodal capabilities, seamlessly integrating screen-based interactions.

In summary, forming corresponding representations with neural networks and connecting them to human hierarchical concepts to address a single symbolic entity represents the initial stage of neuro-symbolic AI. This corresponds to the most basic aspect of human reasoning. Building upon this foundation, if we integrate structures related to the task, it constitutes the intermediate stage of neuro-symbolic AI. The immense workload here lies in constructing numerous classifiers to address individual symbolic entities. Fortunately, current deep pre-trained models are providing a unified base for these smaller classifiers. The crux of issues in areas such as autonomous driving and elderly care robots stems from the lack of integration of this intelligence principle from the base to the top level. They only tackle minor issues which are mentioned in [155], which means that human's common sense level has not been achieved thoroughly.

4.4 The advantages of neuro-symbolic paradigm in AIoT

Neuro-symbolic paradigm, offers several advantages in AIoT applications, particularly in terms of interpretability, testability, and verifiability [138]. Here's how these aspects are enhanced by neuro-symbolic approaches:

Interpretability. Neuro-symbolic techniques enhance the interpretability of models by integrating symbolic reasoning with DL [138, 156]. This combination allows the model to offer explanations based on symbolic logic that are understandable to humans. This is crucial in applications where decisions need to be understandable and justifiable, such as in healthcare or infrastructure management. The integration of symbolic reasoning helps articulate domain-specific knowledge and shows how decisions are derived, making AI decisions more transparent and easier to trust. For example, Daiki et al. [135] proposed a novel neuro-symbolic reinforcement learning method utilizing the Logical Neural Network framework, which is capable of learning symbolic and interpretable rules and integrating these rules into a differentiable network. This approach enhances the interpretability and promotes rapid convergence of policies within text-based games.

Testability. The symbolic component of neuro-symbolic AI allows for more structured and hypothesis-driven testing. Symbolic rules can be explicitly tested against various scenarios [157, 158], which is not straightforward with purely DL models. This explicit structure of symbolic rules enables the testing of specific reasoning paths and intermediate states within the model. Furthermore, it improves the granularity at which models can be tested and ensures that the models behave as expected across a range of conditions and inputs. In AIoT systems, a large number of sensors and devices generate substantial amounts of data, which need to be analyzed by ML models to extract useful information and insights. The neuro-symbolic methods and symbolic regression techniques mentioned in Balla's work [159] can be used to discover and validate models from this data, which is similar to data-driven decision-making in AIoT. They demonstrated how to utilize neuro-symbolic methods to enhance the testability of models in social science research, that is, by discovering and verifying interpretable models from data to generate predictions that can be tested across different populations or periods. The neuro-symbolic methods and symbolic regression techniques discussed in the article can be applied to identify and validate models within this data, which bears resemblance to the data-driven decision-making processes in AIoT.

Verifiability. Neuro-symbolic systems improve verifiability by leveraging the rigor of symbolic logic to ensure that the models adhere to defined logical constraints and rules [156]. This is especially important in safety-critical applications where AI models must operate within strict regulatory and operational frameworks. By using symbolic reasoning, it's possible to verify that the model's outputs comply with these frameworks, providing a layer of safety and compliance that is hard to achieve with models based solely on neural networks. For example, Briti et al. [160] proposed a framework called hierarchical program-triggered reinforcement learning (HPRL), which employs a hierarchical structure to decompose complex autonomous driving tasks into multiple relatively simple sub-tasks. Each sub-task is executed by a trained RL agent that focuses on specific driving strategies, such as driving straight, turning, or changing lanes. Through this hierarchical and program-triggered approach, the HPRL framework not only enhances the interpretability of the RL agents but also strengthens the verifiability of the system through formal verification, which is crucial for the safety of autonomous driving systems.

To summarize, these three characteristics make neuro-symbolic AI particularly suitable for AIoT applications where decisions must not only be accurate but also justifiable, understandable, and compliant with regulatory standards. The ability to explain and verify decisions becomes an advan-

tage in deploying AI in real-world, critical systems where accountability and precision are paramount.

5 RQ3: challenges for neuro-symbolic paradigm

The neuro-symbolic paradigm has shown promise in improving interpretability, verifiability, and testability, thus making AIoT systems more reliable. While it has achieved success in areas like natural language processing and computer vision, its applications in AIoT are still in the early stages. As this technology advances, integrating symbolic reasoning with neural networks will be key to addressing the challenges of IoT data processing and intelligent decision-making. Currently, challenges remain to be addressed for the neuro-symbolic paradigm to fully realize its potential in AIoT.

Manually crafted functions. Neuro-symbolic systems aim to enhance model generalization, enabling them to learn from limited data and adapt to new scenarios. However, achieving this goal is not straightforward. One challenge is integrating symbolic reasoning, which needs to be robust and generalizable across diverse tasks in AIoT, such as different driving scenarios in autonomous driving systems and diverse landing areas for unmanned delivery drones. How can we determine which symbolic rules will generalize well—and which will turn out to be brittle? Such generalizability for symbolic AI is even difficult in relatively simpler applications in terms of data diversity and system complexity. In VQA [124, 130], for example, neuro-symbolic methods aim to combine the strengths of neural networks in processing visual information with symbolic reasoning for answering questions. However, traditional symbolic functions are often handcrafted and designed to work well only within specific contexts or domains. When applied to scenarios unforeseen by their creators, these manually created domain-specific functions may fail to generalize out of the domain they were designed for, leading to reduced performance or inaccuracies in answering questions. One reason for this limitation is the difficulty in designing symbolic functions that can adequately capture the complexity and variability of real-world visual data. Although Suris et al. [137] have employed LLMs, utilized to generate Python code via a provided API for execution, in tackling certain common-sense problems, the core functions are manually created in a domain-specific manner, with no solution offered for the hallucinations LLMs may encounter. Handcrafted rules or heuristics may not be sufficiently flexible to handle the diverse range of questions and images encountered in VQA tasks. Moreover, the reliance on manually crafted symbolic functions can hinder the adaptability and scalability of neuro-symbolic models. As the complexity of VQA tasks increases or as the distribution of visual data shifts, these fixed symbolic functions may become less effective or even obsolete, requiring manual intervention or re-engineering to maintain performance.

Lack of support for multimodal sensor data. As multimodal data (such as images, text, and audio) becomes more prevalent, neuro-symbolic systems need to handle and generalize across these various modalities. This capability requires models to not only comprehend data from individual modalities—but also establish connections across different modalities. Neuro-symbolic models may struggle to generalize to unseen data distributions or adapt to changes in the environment due to the inherent assumption of similarity between the training distribution and real-world data for neural networks [161, 162]. Ensuring robustness to cross-modal distributional shifts is essential for deploying these systems in real-world applications where the data distribution may change over time.

Diverse sensor data are heavily used in AIoT, including LiDAR [163], millimeter-wave radar [164], ultrasonic radar [165], electroencephalography (EEG) [166], electrocardiogram (ECG) [167], accelerometer [165], gyroscope [168], and altimeter [169]. Often, these sensors are used together, and such multimodal sensor data is not yet supported by current neuro-symbolic methods. Handling such multimodality is crucial for the applicability of neuro-symbolic AI in those multi-modal AIoT systems such as autonomous driving cars (cameras, radars, and LiDARs [165]), UAVs (cameras, altimeters, and radars) [170, 171], activity recognition (accelerometers and gyroscopes) [172, 173], and medical human sensors (ECG and EEG) [174]. The complexity of integrating diverse data types and ensuring accurate and robust generalization across these modalities remains a significant challenge. Current neuro-symbolic models need to be enhanced to process and synthesize multimodal data effectively, improving their reliability and performance in real-world AIoT applications [175].

Complex integration of neural networks and symbolic AI. Building integrated neuro-symbolic systems requires expertise in both neural networks and symbolic AI, increasing the development complexity and the need for interdisciplinary collaboration. We illustrate this challenge on two notable approaches that integrate neural networks with symbolic AI: logic tensor networks (LTNs) and DeepProbLog.

LTNs [176] combine DL with first-order logic to perform logical reasoning over data. LTNs use a neural network to embed data into a continuous vector space, where logical formulas are interpreted as differentiable functions. This allows the network to learn the parameters of the logic-based model jointly with the neural network parameters. However, this integration introduces complexity as the logical reasoning process needs to be robust and generalizable across diverse tasks, requiring significant expertise in both neural networks and symbolic reasoning. DeepProbLog [133] is a proba-

bilistic logic programming language that incorporates DL models. In DeepProbLog, neural networks are used to provide probabilistic facts, which are then processed by the logic programming component to perform reasoning. This architecture allows the system to combine the strengths of neural networks in handling raw sensory input with the logical reasoning capabilities of symbolic AI. Despite its advantages, the performance of such integrated systems can be limited by the slow symbolic reasoning processes, which may not match the speed of neural network inference.

Combining neural and symbolic methods often results in complex models that are difficult to scale and debug. The symbolic reasoning components can become a scalability bottleneck, limiting the model's ability to handle large datasets and complex tasks efficiently. The neural part will likely remain difficult to understand and debug for engineers. Ensuring debuggability and scalability in these hybrid models remains a challenge, as they must effectively manage the trade-offs between the flexibility of neural networks and the precision of symbolic reasoning.

Biased interpretability. *Biased interpretability* refers to the bias exhibited by a model during its interpretation process. This bias can originate from training data, model structure, or the reasoning process. (1) Data bias. Bias in the training data affects the model's learning outcomes. If the training data contains more samples of certain categories, the model may be inclined towards these categories, thus displaying bias in its interpretations. (2) Model structure bias. The design of the model may introduce bias. For example, predefined rules in the symbolic reasoning module might favor certain logical paths, leading to biased interpretations. (3) Reasoning process bias. The logical rules and probabilities applied during the reasoning process can introduce bias. For instance, specific reasoning paths might be used more frequently by the model, resulting in biased interpretations. Using NS-CL [136] as an example, NS-CL is a model that combines neural networks and symbolic reasoning, designed for VQA. It uses DL for perception and combines it with symbolic reasoning for interpretable decision-making. Suppose we have a visual question-answering system with the question, "Is there a red ball in the picture?" If the training data contains significantly more images of red balls than balls of other colors, then biased interpretability in NS-CL would manifest in three aspects. (1) Impact of data bias. Since there are more samples of red balls in the training data, the neural network (perception module) might be more likely to recognize ambiguous or unclear balls as red. This bias would transfer to the symbolic reasoning module, leading the system to interpret the presence of a red ball, even in cases where there might not actually be one in the image. (2) Impact of model structure bias. If the rules defined in the symbolic reasoning module favor identifying color over shape (e.g., prioritizing color judgment), the model would tend to confirm the color first when answering questions. This structural bias would result in the model emphasizing color features in its interpretation while possibly neglecting subtle shape differences. (3) Impact of reasoning process bias: During the reasoning process, if the model more frequently applies certain logical rules (e.g., classifying based on color), it would rely more on these rules when interpreting results. Consequently, this would lead to a biased interpretation process. Through the example of NS-CL, we can clearly see how "biased interpretability" manifests in data, model structure, and the reasoning process. Although neuro-symbolic models aim to combine the strengths of neural networks and symbolic reasoning to provide interpretable decision-making processes, bias can still be introduced at various stages. To mitigate these biases, improvements are needed in data collection, model design, and the reasoning process.

Trade-offs between interpretability and performance. The difficulty in balancing interpretability and performance in neuro-symbolic models stems from the fact that improving the model's interpretability typically requires sacrificing a certain degree of performance, and vice versa [130]. Again, use NS-CL [136] as an example. Using a purely DL model, such as a deep CNN for VQA, can achieve high performance on benchmarks because it can learn complex patterns from data. However, the decision-making process of the model is a black box, making it difficult to understand why certain decisions are made. Using a purely symbolic AI system provides high interpretability because every step of the reasoning process is transparent and based on predefined rules. However, such systems often struggle with raw perceptual data and complex pattern recognition, leading to lower performance in tasks requiring detailed image understanding. NS-CL seeks to balance the strengths of neural networks and symbolic reasoning. The neural component (CNN) handles complex perception tasks, while the symbolic component ensures the interpretability of the reasoning process. While NS-CL offers better interpretability compared to purely neural models, its performance may not reach that of state-of-the-art DL models. Additionally, integrating these two components can introduce complexity, affecting overall system efficiency. NS-CL illustrates the trade-off between interpretability and performance. By combining neural and symbolic approaches, it aims to leverage the strengths of both, providing a more interpretable decision-making process while maintaining reasonable performance. However, achieving the optimal balance is challenging because improving one aspect often impacts the other.

Meanwhile, quantifying and verifying the model's interpretability capabilities remains a challenge [177]. Interpretability may be subjective, depending on the perspectives and needs of stakeholders: what one person considers interpretable may not be so for another. Neuro-symbolic models typically include components based on neural networks and

symbolic reasoning modules, leading to complex interactions and decision-making processes that are difficult to interpret. Despite efforts to integrate symbolic reasoning, many neuro-symbolic models still exhibit black-box behavior that is hard to understand. Additionally, the lack of standardized evaluation metrics to quantify interpretability makes it difficult to compare different methods objectively.

Ethical and societal challenges. The application of neuro-symbolic paradigms spans various fields, from autonomous driving to medical diagnosis, bringing forth a broad range of ethical and societal challenges in these domains. Firstly, neuro-symbolic models typically require large amounts of data for training and optimization. Various types of data are used, including structured and unstructured data, which may involve personal privacy information such as medical records and financial details [157]. In neuro-symbolic approaches, models may learn patterns related to personal identities or sensitive information [178], posing risks of information leakage. Even if personal identity information is not explicitly used during model training, the model may still indirectly infer personal identities through learned correlated information [179]. Neuro-symbolic methods may be used to analyze and mine large-scale data [133], leading to risks of data misuse. Incorrect usage or analysis of such data can have adverse effects on personal privacy and societal fairness. Additionally, the computational systems and algorithms used in neuro-symbolic methods may have security vulnerabilities [180], allowing malicious users to exploit these vulnerabilities to access sensitive information or manipulate model behavior, thereby increasing the risks of privacy breaches and data misuse.

Secondly, if there is bias or discrimination present in the training data, neuro-symbolic systems may learn these biases and reflect them in their reasoning and decision-making processes [157, 181]. This means that the model may make unfair decisions based on biases present in the training data, resulting in outcomes biased towards certain groups or holding discriminatory attitudes towards certain groups, thereby affecting the fairness and reliability of the model. Consequently, the application of neuro-symbolic models in the real world may have negative societal impacts, exacerbating social inequalities and causing further harm and exclusion to affected groups. When models demonstrate bias or discrimination, people may become suspicious and distrustful of them, which reduces their acceptability and applicability. This could lead to reluctance to use or rely on these models, thereby limiting their actual application and impact.

Furthermore, when neuro-symbolic systems encounter errors or produce adverse effects, determining responsibility and accountability becomes a complex issue [182]. Neuro-symbolic models typically consist of multiple components, including neural network parts and symbolic reasoning parts. This complexity makes it difficult to determine the responsibility for specific issues or errors because problems may involve interactions and influences between different components. Due to the system's complexity, it is challenging to pinpoint which component or data point caused the problem. Neuro-symbolic methods combine statistical learning from neural networks with logical reasoning from symbolic inference, so the model may use both methods interchangeably during the reasoning process. This mixture of reasoning makes it difficult to determine whether the model's decisions in specific situations are based on statistical learning or logical reasoning, thereby complicating the determination of responsibility and accountability. Moreover, the neural network component is typically a black-box model, making its internal mechanisms and decision-making processes difficult to understand and explain. Thus, it is challenging to determine whether the model's behavior in specific situations is reasonable and why errors or adverse outcomes occur. Additionally, neuro-symbolic methods are often data-driven, meaning their behavior and decisions are often influenced by the training data. Therefore, when the model encounters problems or errors, responsibility may partially be attributed to the quality and biases of the training data, rather than solely to the design or implementation of the model itself.

Lastly, incorporating ethical reasoning into neuro-symbolic methods also presents challenges [183], especially in situations where decisions affect individuals or groups, as morality is subjective and context-dependent. Moral reasoning involves various aspects, including ethical principles, values, and emotional factors, which are difficult to integrate in a unified manner within neuro-symbolic methods. Due to the complexity of ethical reasoning, models often struggle to accurately understand and infer human moral decision-making processes. Different cultures, groups, and individuals may have varying ethical standards and values, adding to the difficulty of incorporating ethical reasoning in neuro-symbolic methods. Models need to consider this diversity and account for the influence of different ethical standards during the reasoning process. Additionally, the black-box nature of the neural network component in neuro-symbolic methods makes the model's ethical reasoning process difficult to understand and scrutinize, thereby increasing uncertainty and risk. The application of neuro-symbolic methods may have significant societal impacts and consequences, particularly in fields involving ethical decision-making, such as healthcare [184], justice [185], and military [186]. Therefore, models need to consider the societal impacts and consequences when incorporating ethical reasoning and take appropriate measures to ensure the fairness and rationality of ethical decision-making.

In summary, the above challenges highlight the obstacles that neuro-symbolic approaches need to overcome in achieving their objectives, while also providing directions for future research. With advancements in learning-enabled IoT tech-

nology and increased research efforts, these challenges are likely to be addressed, thereby driving the further development of the neuro-symbolic field.

6 Conclusion

The integration of neuro-symbolic approaches within the AIoT presents a stride towards creating intelligent systems capable of sophisticated reasoning and decision-making. This paper has undertaken a comprehensive review of the current state-of-the-art DL models, highlighting their limitations in terms of testability, verifiability, and interpretability. We have underscored the potential of neuro-symbolic methods to transcend these limitations by synergizing the robustness of symbolic AI with the adaptability of DL. The survey has delineated the progress and prospects of neuro-symbolic AI across various domains, including autonomous driving, service robots, and systems with highly limited data. We have accentuated the unique advantages that the neuro-symbolic paradigm offers in enhancing the interpretability, testability, and verifiability of AIoT systems. These advantages are pivotal for the deployment of AI in safety-critical applications where decisions must be not only accurate but also justifiable and transparent. However, the journey towards the seamless integration of neuro-symbolic AI in AIoT systems is fraught with challenges. We have identified and discussed significant hurdles such as hard-coded symbolic AI, lack of support for multimodal sensor data, complex integration of neural networks and symbolic AI, biased interpretability, trade-offs between interpretability and performance, and ethical and societal challenges. Addressing these challenges calls for concerted efforts in research and development, interdisciplinary collaboration, and the formulation of standardized evaluation metrics. The ethical and societal implications of neuro-symbolic AI are profound, extending from data privacy concerns to the fairness and accountability of AI systems. As we advance, it is imperative to integrate ethical reasoning into neuro-symbolic methods, ensuring that AI systems respect human values and societal norms. In conclusion, the neuro-symbolic approach holds significant potential for advancing AIoT by improving the reliability, interpretability, and decision-making of intelligent systems. The path forward involves overcoming the existing challenges, fostering interdisciplinary research, and developing ethical guidelines that will usher in a new generation of AIoT applications that are not only technologically advanced but also socially responsible and trustworthy. As we look ahead, the future of neuro-symbolic AI in AIoT is bright but requires careful navigation through the complex landscape of technological and ethical considerations. With dedicated research efforts and a commitment to addressing the challenges identified in this survey, we can pave the way for a new era of AIoT that is underpinned by the powerful and interpretable neuro-symbolic AI.

Author Contributions Z.L. and I.A. wrote the main manuscript text. H.J.K., I.R., and X.Z. provided ideas and key papers to aid the understanding of the two lead students and offered guidance throughout the writing process. They also rewrote key subsections, reviewed the entire manuscript, and provided constructive comments for overall improvement.

Funding Open Access funding enabled and organized by CAUL and its Member Institutions

Data Availability No datasets were generated or analysed during the current study.

Declarations

Conflict of interest The authors declare no competing interests.

References

1. Feifei S et al (2020) Recent progress on the convergence of the internet of things and artificial intelligence. IEEE Netw 34(5):8–15. https://doi.org/10.1109/MNET.011.2000009
2. Tharindu Bandaragoda A et al (2020) Artificial intelligence based commuter behaviour profiling framework using internet of things for real-time decision-making. Neural Comput Appl 32:16057–16071. https://doi.org/10.1007/s00521-020-04736-7
3. McKinsey & Company (2013) Disruptive technologies: Advances that will transform life, business, and the global economy. https://www.mckinsey.com/mgi
4. Hu M, Liu W, Peng K et al (2018) Joint routing and scheduling for vehicle-assisted multidrone surveillance. Internet Things J 6(2):1781–1790. https://doi.org/10.1109/JIOT.2018.2878602
5. Cai C, Hu M, Cao D et al (2019) Self-deployable indoor localization with acoustic-enabled IoT devices exploiting participatory sensing. Internet Things J 6(3):5297–5311. https://doi.org/10.1109/JIOT.2019.2900524
6. Waymo: Autonomous Car. https://waymo.com/. Accessed: 2024-04-29
7. Abate F, Carratù M, Liguori C et al (2018) Smart meter for the IoT. In: I2MTC, pp. 1–6. https://doi.org/10.1109/I2MTC.2018.8409838
8. Zhang S, Callaghan V, An X et al (2022) Feature selection and human arm activity classification using a wristband. J Reliab Intell Environ 8(3):285–298. https://doi.org/10.1007/s40860-022-00181-6

9. Al-Sakran HO et al (2015) Intelligent traffic information system based on integration of internet of things and agent technology. IJACSA 6(2):37–43
10. Douiba M, Benkirane S, Guezzaz A et al (2023) Anomaly detection model based on gradient boosting and decision tree for iot environments security. J Reliab Intell Environ 9(4):421–432. https://doi.org/10.1007/s40860-022-00184-3
11. Nagarajan SM, Muthukumaran V, Murugesan R et al (2022) Innovative feature selection and classification model for heart disease prediction. J Reliab Intell Environ 8(4):333–343. https://doi.org/10.1007/s40860-021-00152-3
12. Chen T, Madanian S, Airehrour D et al (2022) Machine learning methods for hospital readmission prediction: systematic analysis of literature. J Reliab Intell Environ 8(1):49–66. https://doi.org/10.1007/s40860-021-00165-y
13. Kaur J, Khehra BS, Singh A (2023) Back propagation artificial neural network for diagnose of the heart disease. J Reliab Intell Environ 9(1):57–85. https://doi.org/10.1007/s40860-022-00192-3
14. Hosseini S, Sardo SR (2023) Network intrusion detection based on deep learning method in internet of thing. J Reliab Intell Environ 9(2):147–159. https://doi.org/10.1007/s40860-021-00169-8
15. Xiaowei H, Daniel K, Wenjie R et al (2020) A survey of safety and trustworthiness of deep neural networks: verification, testing, adversarial attack and defence, and interpretability. Comput Sci Rev 37:100270. https://doi.org/10.1016/j.cosrev.2020.100270
16. Jie MZ, Mark H, Lei M, Yang L (2020) Machine learning testing: survey, landscapes and horizons. Trans Softw Eng 48(1):1–36. https://doi.org/10.1109/TSE.2019.2962027
17. Xiang W, Musau P, Wild AA et al (2018) Verification for machine learning, autonomy, and neural networks survey. https://doi.org/10.48550/arXiv.1810.01989
18. Supriyo C, Richard et al (2017) Interpretability of deep learning models: a survey of results. In: IEEE SmartWorld, pp 1–6. https://doi.org/10.1109/UIC-ATC.2017.8397411
19. Zhang J, Li J (2020) Testing and verification of neural-network-based safety-critical control software: a systematic literature review. Inf Softw Technol 123:106296. https://doi.org/10.1016/j.infsof.2020.106296
20. Howard AG, Zhu M, Chen B et al (2017) Mobilenets: efficient convolutional neural networks for mobile vision applications. https://doi.org/10.48550/arXiv.1704.04861
21. Iandola FN, Han S, Moskewicz MW et al (2016) Squeezenet: Alexnet-level accuracy with 50x fewer parameters and <0.5 mb model size. https://doi.org/10.48550/arXiv.1602.07360
22. Szegedy C, Liu W et al (2015) Going deeper with convolutions. In: CVPR, pp 1–9 https://doi.org/10.1109/CVPR.2015.7298594
23. Rodrigues DdA, Ivo RF, Satapathy SC et al (2020) A new approach for classification skin lesion based on transfer learning, deep learning, and IoT system. Pattern Recogn Lett 136:8–15. https://doi.org/10.1016/j.patrec.2020.05.019
24. Taiwo O, Ezugwu AE (2021) Internet of things-based intelligent smart home control system. Secur Commun Netw. https://doi.org/10.1155/2021/9928254
25. Bianchi V, Bassoli M, Lombardo G et al (2019) IoT wearable sensor and deep learning: an integrated approach for personalized human activity recognition in a smart home environment. Internet Things J 6(5):8553–8562. https://doi.org/10.1109/JIOT.2019.2920283
26. Magaia N, Fonseca R, Muhammad K et al (2020) Industrial internet-of-things security enhanced with deep learning approaches for smart cities. Internet Things J 8(8):6393–6405. https://doi.org/10.1109/JIOT.2020.3042174
27. Xin Q, Alazab M et al (2022) A deep learning architecture for power management in smart cities. Energy Rep 8:1568–1577. https://doi.org/10.1016/j.egyr.2021.12.053
28. Jeon Ahmed I et al (2022) A smart IoT enabled end-to-end 3D object detection system for autonomous vehicles. TITS. https://doi.org/10.1109/TITS.2022.3210490
29. Shon T (2021) In-vehicle networking/autonomous vehicle security for internet of things/vehicles. Electronics. https://doi.org/10.3390/electronics10060637
30. Mehra M, Saxena S, Sankaranarayanan S et al (2018) IoT based hydroponics system using deep neural networks. Comput Electron Agric 155:473–486. https://doi.org/10.1016/j.compag.2018.10.015
31. Zhang X (2021) Application of human motion recognition utilizing deep learning and smart wearable device in sports. IJSAEM 12(4):835–843. https://doi.org/10.1007/s13198-021-01118-7
32. Vimal S, Robinson YH, Kadry S et al (2021) IoT based smart health monitoring with CNN using edge computing. J Internet Technol 22(1):173–185
33. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735
34. Cho K, Merriënboer V et al (2024) Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078
35. Schuster M, Paliwal KK (1997) Bidirectional recurrent neural networks. TSP 45(11):2673–2681. https://doi.org/10.1109/78.650093
36. Li S, Li W, Cook C et al (2018) Independently recurrent neural network (INDRNN): Building a longer and deeper rnn. In: CVPR, pp 5457–5466. https://doi.org/10.1109/CVPR.2018.00572
37. Yu M, Xu F et al (2021) Using long short-term memory (LSTM) and internet of things (IoT) for localized surface temperature forecasting in an urban environment. IEEE Access 9:137406–137418. https://doi.org/10.1109/ACCESS.2021.3116809
38. Ullah I, Mahmoud QH (2022) Design and development of RNN anomaly detection model for iot networks. IEEE Access 10:62722–62750. https://doi.org/10.1109/ACCESS.2022.3176317
39. Farsi M (2021) Application of ensemble rnn deep neural network to the fall detection through iot environment. Alex Eng J 60(1):199–211. https://doi.org/10.1016/j.aej.2020.06.056
40. Saravanan D, Kumar KS (2023) Improving air pollution detection accuracy and quality monitoring based on bidirectional rnn and the internet of things. Mater Today Proc 81:791–796. https://doi.org/10.1016/j.matpr.2021.04.239
41. Abdellah AR, Koucheryavy A (2020) Deep learning with long short-term memory for IOT traffic prediction. In: NEW2AN. Springer, pp 267–280. https://doi.org/10.1007/978-3-030-65726-0_24
42. Sharma M, Kaur P (2023) XLAAM: explainable LSTM-based activity and anomaly monitoring in a fog environment. J Reliab Intell Environ 9(4):463–477. https://doi.org/10.1007/s40860-022-00185-2
43. Vlahogianni EI, Kepaptsoglou K et al (2016) A real-time parking prediction system for smart cities. J Intell Transp Syst 20(2):192–204. https://doi.org/10.1080/15472450.2015.1037955
44. Alladi T, Kohli V et al (2023) A deep learning based misbehavior classification scheme for intrusion detection in cooperative intelligent transportation systems. Digit Commun Netw 9(5):1113–1122. https://doi.org/10.1016/j.dcan.2022.06.018
45. Alkahtani H, Aldhyani TH (2021) Botnet attack detection by using CNN-LSTM model for internet of things applications. Secur Commun Netw 2021:1–23. https://doi.org/10.1155/2021/3806459
46. Hinton GE (2012) In: Montavon G, Orr GB, Müller K-R (eds) A practical guide to training restricted boltzmann machines, pp 599–619. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-35289-8_32

47. Hinton GE, Osindero S, Teh Y-W (2006) A fast learning algorithm for deep belief nets. Neural Comput 18(7):1527–1554. https://doi.org/10.1162/neco.2006.18.7.1527
48. Peter Soosai Anandaraj A, Gomathy V et al (2021) Internet of medical things (IoMT) enabled skin lesion detection and classification using optimal segmentation and restricted Boltzmann machines. Cogn Internet Med Things Smart Healthc Serv Appl. https://doi.org/10.1007/978-3-030-55833-8_12
49. Ma X, Yu H et al (2015) Large-scale transportation network congestion evolution prediction using deep learning theory. PLoS One 10(3):0119044. https://doi.org/10.1371/journal.pone.0119044
50. Elsaeidy A, Munasinghe KS et al (2019) Intrusion detection in smart cities using restricted Boltzmann machines. J Netw Comput Appl 135:76–83. https://doi.org/10.1016/j.jnca.2019.02.026
51. Ng A et al (2011) Sparse autoencoder. CS294A Lect Notes 72:1–19
52. Vincent P, Larochelle H et al (2010) Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion. J Mach Learn Res 10(5555/1756006):1953039
53. Kingma DP, Welling M (2013) Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114
54. Masci J, Meier U et al (2011) Stacked convolutional auto-encoders for hierarchical feature extraction. In: ICANN. Springer, pp 52–59. https://doi.org/10.1007/978-3-642-21735-7_7
55. Yin C, Zhang S et al (2020) Anomaly detection based on convolutional recurrent autoencoder for iot time series. TSMC 52(1):112–122. https://doi.org/10.1109/TSMC.2020.2968516
56. Chen J, Li T et al (2020) WSN sampling optimization for signal reconstruction using spatiotemporal autoencoder. IEEE Sens J 20(23):14290–14301. https://doi.org/10.1109/JSEN.2020.3007369
57. Himeur Y, Alsalemi A et al (2021) Detection of appliance-level abnormal energy consumption in buildings using autoencoders and micro-moments. In: International conference on big data internet things. Springer, pp 179–193. https://doi.org/10.1007/978-3-031-07969-6_14
58. Yu T (2018) UAV-enabled spatial data sampling in large-scale IoT systems using denoising autoencoder neural network. IEEE IoT 6(2):1856–1865. https://doi.org/10.1109/JIOT.2018.2876695
59. Ashraf J, Bakhshi AD et al (2020) Novel deep learning-enabled LSTM autoencoder architecture for discovering anomalous events from intelligent transportation systems. TITS 22(7):4507–4518. https://doi.org/10.1109/TITS.2020.3017882
60. Zhang S, Yao Y et al (2019) Deep autoencoder neural networks for short-term traffic congestion prediction of transportation networks. Sensors 19(10):2229. https://doi.org/10.3390/s19102229
61. Bae G, Jang S et al (2019) Autoencoder-based on anomaly detection with intrusion scoring for smart factory environments. In: PDCAT. Springer, pp 414–423. https://doi.org/10.1007/978-981-13-5907-1_44
62. Kipf TN, Welling M (2016) Semi-supervised classification with graph convolutional networks.https://doi.org/10.48550/arXiv.1609.02907
63. Kipf TN, Welling M (2016) Variational graph auto-encoders. https://doi.org/10.48550/arXiv.1611.07308
64. Hajiramezanali E, Hasanzadeh A et al. (2019) Variational graph recurrent neural networks. NeurIPS https://doi.org/10.48550/arXiv.1908.09710
65. Sheng Z, Xu Y et al (2022) Graph-based spatial-temporal convolutional network for vehicle trajectory prediction in autonomous driving. TITS 23(10):17654–17665. https://doi.org/10.1109/TITS.2022.3155749
66. Karimi AM, Wu Y et al (2021) Spatiotemporal graph neural network for performance prediction of photovoltaic power systems. AAAI 35:15323–15330. https://doi.org/10.1609/aaai.v35i17.17799
67. Rangesh A, Maheshwari et al (2021) TrackMPNN: a message passing graph neural architecture for multi-object tracking. https://doi.org/10.48550/arXiv.2101.04206
68. Gama F, Tolstaya et al. (2021) Graph neural networks for decentralized controllers. In: ICASSP, pp 5260–5264. https://doi.org/10.1109/ICASSP39728.2021.9414563
69. Lin D, Lin J, Zhao L et al (2021) Multilabel aerial image classification with a concept attention graph neural network. TGRS 60:1–12. https://doi.org/10.1109/TGRS.2020.3041461
70. Bi Y, Chadha A, Abbas et al (2019) Graph-based object classification for neuromorphic vision sensing. In: ICCV, pp 491–501. https://doi.org/10.1109/ICCV.2019.00058
71. Jin G, Wang M et al (2022) STGNN-TTE: travel time estimation via spatial-temporal graph neural network. Future Gener Comput Syst 126:70–81. https://doi.org/10.1016/j.future.2021.07.012
72. Mondal R (2020) A new framework for smartphone sensor-based human activity recognition using graph neural network. IEEE Sens J 21(10):11461–11468. https://doi.org/10.1109/JSEN.2020.3015726
73. Guo C, Zhong Z et al (2022) Neurstrucenergy: a bi-directional GNN model for energy prediction of neural networks in IoT. DCN. https://doi.org/10.1016/j.dcan.2022.09.006
74. Bronstein M et al (2021) Geometric deep learning: grids, groups, graphs, geodesics, and gauges. https://doi.org/10.48550/arXiv.2104.13478
75. Sun P, Kretzschmar et al (2020) Scalability in perception for autonomous driving: Waymo open dataset. In: CVPR, pp 2446–2454. https://doi.org/10.1109/CVPR42600.2020.00252
76. Nouri A, Charrier et al (2017) Technical report: Greyc 3D colored mesh database. Ph.D. thesis, Normandie Université, Unicaen, EnsiCaen, CNRS, GREYC UMR 6072
77. Fei Y et al (2023) A survey of geometric optimization for deep learning: from Euclidean space to Riemannian manifold. https://doi.org/10.48550/arXiv.2302.08210
78. Edelsbrunner H et al (2022) Computational topology: an introduction. American Mathematical Society, Providence
79. Guggenheimer HW (2012) Differential geometry. Courier Corporation, New York
80. West DB et al (2001) Introduction to graph theory, vol 2. Prentice Hall, Upper Saddle River
81. Ju C, Guan C (2022) Tensor-cspnet: a novel geometric deep learning framework for motor imagery classification. TNNLS. https://doi.org/10.1109/TNNLS.2022.3172108
82. Papakis I, Sarkar et al (2021) A graph convolutional neural network based approach for traffic monitoring using augmented detections with optical flow. In: ITSC, pp 2980–2986. https://doi.org/10.1109/ITSC48978.2021.9564655
83. Villalba-Díez J, Molina M et al (2020) Geometric deep lean learning Deep learning in industry 4.0 cyber-physical complex networks. Sensors 20(3):763. https://doi.org/10.3390/s20030763
84. Qin C, Srivastava AK et al (2023) Geometric deep-learning-based spatiotemporal forecasting for inverter-based solar power. Syst J. https://doi.org/10.1109/JSYST.2023.3250403
85. Atz K, Grisoni F, Schneider G (2021) Geometric deep learning on molecular representations. Nat Mach Intell 3(12):1023–1032. https://doi.org/10.1038/s42256-021-00418-8
86. Lu J, Tian Y, Zhang Y et al (2023) LGL-BCI: A lightweight geometric learning framework for motor imagery-based brain-computer interfaces. arXiv preprint arXiv:2310.08051
87. James J (2021) Citywide traffic speed prediction: a geometric deep learning approach. Knowl Based Syst 212:106592. https://doi.org/10.1016/j.knosys.2020.106592

88. Monti F, Boscaini et al (2017) Geometric deep learning on graphs and manifolds using mixture model cnns. In: CVPR, pp 5115–5124. https://doi.org/10.1109/CVPR.2017.576
89. Huang Z, Van Gool L (2017) A riemannian network for spd matrix learning. In: AAAI, 31. https://doi.org/10.1609/aaai.v31i1.10866
90. Ashish V, Noam S et al. (2017) Attention is all you need. NeurIPS https://doi.org/10.48550/arXiv.1706.03762
91. Shao T, Guo Y et al (2019) Transformer-based neural network for answer selection in question answering. IEEE Access 7:26146–26156. https://doi.org/10.1109/ACCESS.2019.2900753
92. Devlin J, Chang et al (2018) Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805
93. Liu Y, Ott M et al (2019) Roberta: a robustly optimized bert pretraining approach. https://doi.org/10.48550/arXiv.1907.11692
94. Simunec M, Soic R (2023) Smart home notifications in Croatian language: a transformer-based approach. In: ConTEL, pp 1–5. https://doi.org/10.1109/ConTEL58387.2023.10199029
95. Luo Y, Chen X et al (2022) Transformer-based device-type identification in heterogeneous IoT traffic. IEEE IoT 10(6):5050–5062. https://doi.org/10.1109/JIOT.2022.3221967
96. Rashvand N, Witham K et al (2024) Enhancing automatic modulation recognition for IoT applications using transformers. IoT 5(2):212–226. https://doi.org/10.3390/iot5020011
97. Chen Z, Chen D et al (2021) Learning graph structures with transformer for multivariate time-series anomaly detection in IoT. IEEE IoT 9(12):9179–9189. https://doi.org/10.1109/JIOT.2021.3100509
98. Ullah S, Ahmad J et al (2023) TNN-IDS: transformer neural network-based intrusion detection system for MQTT-enabled IoT networks. Comput Netw 237:110072. https://doi.org/10.1016/j.comnet.2023.110072
99. Hu X, Chu L, Pei J et al (2021) Model complexity of deep learning: a survey. Knowl Inf Syst 63:2585–2619. https://doi.org/10.1007/s10115-021-01605-0
100. Mayer R, Jacobsen H-A (2020) Scalable deep learning on distributed infrastructures: challenges, techniques, and tools. CSUR 53(1):1–37. https://doi.org/10.1145/3363554
101. Han J, Cen et al (2024) A survey of geometric graph neural networks: data structures, models and applications. https://doi.org/10.48550/arXiv.2403.00485
102. Yan F, Ruwase et al (2015) Performance modeling and scalability optimization of distributed deep learning systems. In: KDD, pp 1355–1364. https://doi.org/10.1145/2783258.2783270
103. Xi Z, Mok AK et al (2024) Testing learning-enabled cyber-physical systems with large-language models: a formal approach. https://doi.org/10.48550/arXiv.2311.07377
104. Pei K, Cao Y, Suman (2017) Deepxplore: automated whitebox testing of deep learning systems. In: SOSP, pp 1–18. https://doi.org/10.1145/3132747.3132785
105. Geyer J, Kassahun et al (2020) A2d2: audi autonomous driving dataset. https://doi.org/10.48550/arXiv.2004.06320
106. Harel-Canada F, Wang et al (2020) Is neuron coverage a meaningful measure for testing deep neural networks? In: ESEC/FSE, pp 851–862. https://doi.org/10.1145/3368089.3409754
107. Deng Y, Zheng X et al (2022) A declarative metamorphic testing framework for autonomous driving. TSE. https://doi.org/10.1109/TSE.2022.3206427
108. Tuncali CE et al (2019) Requirements-driven test generation for autonomous vehicles with machine learning components. TIV 5(2):265–280. https://doi.org/10.1109/TIV.2019.2955903
109. Qin X, Aréchiga N et al (2023) Robust testing for cyber-physical systems using reinforcement learning. In: MEMOCODE, pp 36–46. https://doi.org/10.1145/3610579.3611087
110. Li X, Xiong H et al (2022) Interpretable deep learning: Interpretation, interpretability, trustworthiness, and beyond. Knowl Inf Syst 64(12):3197–3234. https://doi.org/10.1007/s10115-022-01756-8
111. Rudin C et al (2022) Interpretable machine learning: fundamental principles and 10 grand challenges. Stat Surv 16:1–85. https://doi.org/10.1214/21-SS133
112. Gosiewska A, Kozak A et al (2021) Simpler is better: lifting interpretability-performance trade-off via automated feature engineering. Decis Support Syst 150:113556. https://doi.org/10.1016/j.dss.2021.113556
113. Kucklick J-P, Müller O (2023) Tackling the accuracy-interpretability trade-off: interpretable deep learning models for satellite image-based real estate appraisal. ACM Trans Manag Inf Syst 14(1):1–24. https://doi.org/10.1145/3567430
114. Olah C, Schubert L et al (2017) Feature visualization. Distill. https://doi.org/10.23915/distill.00007
115. Bach S, Binder A et al (2015) On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. PLoS One 10(7):0130140. https://doi.org/10.1371/journal.pone.0130140
116. Umbrello S, Yampolskiy RV (2022) Designing AI for explainability and verifiability: a value sensitive design approach to avoid artificial stupidity in autonomous vehicles. IJSR 14(2):313–322. https://doi.org/10.1007/s12369-021-00790-w
117. Patil KR, Heinrichs B (2022) Verifiability as a complement to AI explainability: A conceptual proposal. https://philsci-archive.pitt.edu/20297/
118. Wang S, Zhang H, Xu K et al (2021) Beta-crown: efficient bound propagation with per-neuron split constraints for neural network robustness verification. NeurIPS 34:29909–29921
119. Feng J, Chai Y, Xu C (2021) A novel neural network to nonlinear complex-variable constrained nonconvex optimization. J Frankl Inst 358(8):4435–4457. https://doi.org/10.1016/j.jfranklin.2021.02.029
120. Danilova M, Dvurechensky P et al (2021) Recent theoretical advances in non-convex optimization. Springer, Berlin. https://doi.org/10.1007/978-3-031-00832-0_3
121. He Z, Zhang T, Lee RB (2018) VeriDeep: verifying integrity of deep neural networks through sensitive-sample fingerprinting.https://doi.org/10.48550/arXiv.1808.03277
122. Xiang L, Zeng X, Wu S et al (2021) Computation of CNN's sensitivity to input perturbation. Neural Process Lett 53:535–560. https://doi.org/10.1007/s11063-020-10420-7
123. Deng Y, Zheng X, Zhang T et al (2020) An analysis of adversarial attacks and defenses on autonomous driving models. In: PerCom, pp 1–10. https://doi.org/10.1109/PerCom45495.2020.9127389
124. Hu R, Andreas J, Rohrbach M et al (2017) Learning to reason: end-to-end module networks for visual question answering. In: ICCV, pp 804–813. https://doi.org/10.1109/ICCV.2017.93
125. Towell GG, Shavlik JW, Noordewier MO (1990) Refinement of approximate domain theories by knowledge-based neural networks. In: AAAI
126. Garcez ASA, Zaverucha G (1999) The connectionist inductive learning and logic programming system. Applied Intelligence 11:59–77
127. Jaeger H (2017) Controlling recurrent neural networks by conceptors. arXiv:1403.3369
128. Graves A, Wayne G, Danihelka I (2014) Neural turing machines. arXiv preprint arXiv:1410.5401
129. Sainbayar S, Arthur S, Jason W et al (2015) End-to-end memory networks. In: NIPS. MIT Press, Cambridge, MA, USA, pp 2440–2448
130. Yi K, Gan C, Li Y, Kohli P (2020) CLEVRER: collision events for video representation and reasoning. arXiv:1910.01442
131. Wickramarachchi R, Henson C, Sheth A (2020) An evaluation of knowledge graph embeddings for autonomous driving data: experience and practice. https://doi.org/10.48550/arXiv.2003.00344

132. Liu Z, Wang Z, Lin Y, Li H (2022) A neural-symbolic approach to natural language understanding. https://doi.org/10.48550/arXiv.2203.10557
133. Manhaeve R, Dumancic S, Kimmig A et al (2018) Deepproblog: neural probabilistic logic programming. Adv Neural Inf Process Syst 31:1–14
134. Parisotto E, Mohamed A-R, Singh R et al (2016) Neuro-symbolic program synthesis. https://doi.org/10.48550/arXiv.1611.01855
135. Kimura D, Ono M, Chaudhury S, et al. (2021) Neuro-symbolic reinforcement learning with first-order logic. https://doi.org/10.48550/arXiv.2110.10963
136. Iayuan M, Chuang G, Pushmeet K et al (2019) The neuro-symbolic concept learner: interpreting scenes, words, and sentences from natural supervision. arXiv:1904.12584
137. Surís D, Menon S, Vondrick C (2023) ViperGPT: visual inference via python execution for reasoning. arXiv:2303.08128
138. Djallel B, Aggarwal CC (2022) Survey on applications of neurosymbolic artificial intelligence. arXiv:2209.12618
139. Bulla C, Birje MN (2022) Improved data-driven root cause analysis in fog computing environment. J Reliab Intell Environ 8(4):359–377
140. Raedt LD, Kimmig A, Toivonen H (2007) ProbLog: a probabilistic prolog and its application in link discovery. IJCAI. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp 2468–2473
141. Hu X, Li G, Liu F, Jin Z (2019) Program generation and code completion techniques based on deep learning: literature review. J Softw 30(5):1206–1223
142. Wooldridge M, Jennings NR (1995) Intelligent agents: theory and practice. Knowl Eng Rev 10(2):115–152
143. Schmidtke HR (2018) A survey on verification strategies for intelligent transportation systems. J Reliab Intell Environ 4(4):211–224
144. Andreas J, Rohrbach M, Darrell T, Klein D (2017) Neural module networks. https://doi.org/10.48550/arXiv.1511.02799
145. Gilpin LH, Ilievski F (2021) Neuro-symbolic reasoning in the traffic domain. J AI Res 15(3):123–145. https://doi.org/10.1234/jair.v15i3.567
146. Sun J, Sun H, Han T, Zhou B (2020) Neuro-symbolic program search for autonomous driving decision module design. In: CoRL
147. Sharifi I, Yildirim M, Fallah S (2023) Towards safe autonomous driving policies using a neuro-symbolic deep reinforcement learning approach. arXiv:2307.01316
148. Mahon BZ, Hickok G (2016) Arguments about the nature of concepts: symbols, embodiment, and beyond. Psychon Bull Rev 23:941–958. https://doi.org/10.3758/s13423-016-1045-2
149. Kalithasan N, Singh H, Bindal V et al (2023) Learning neuro-symbolic programs for language guided robot manipulation. In: ICRA, pp 7973–7980. https://doi.org/10.1109/ICRA48891.2023.10160545
150. Gubbi S, Venkatesh Biswas A et al (2021) Spatial reasoning from natural language instructions for robot manipulation. In: ICRA, pp 11196–11202. https://doi.org/10.1109/ICRA48506.2021.9560895
151. Hanson D, Imran A, Vellanki A, Kanagaraj S (2020) A neuro-symbolic humanlike arm controller for Sophia the robot
152. Carraro T (2023) Overcoming recommendation limitations with neuro-symbolic integration. In: RecSys. RecSys '23. Association for Computing Machinery, New York, NY, USA, pp 1325–1331. https://doi.org/10.1145/3604915.3608876
153. Shakya A, Rus V, Venugopal D (2021) Student strategy prediction using a neuro-symbolic approach. ERIC Number: ED615630
154. Fan Y, Bowden KK et al (2023) Athena 3.0: personalized multimodal chatbot with neuro-symbolic dialogue generators. Alexa Prize Soc Bot Grand Challenge 5. Amazon Science
155. Yu D, Yang B, Liu D, Wang H, Pan S (2023) A survey on neural-symbolic learning systems. Neural Netw 166:105–126. https://doi.org/10.1016/j.neunet.2023.06.028
156. Rawat DB (2023) Towards neuro-symbolic AI for assured and trustworthy human-autonomy teaming. In: TPS-ISA, Los Alamitos, CA, USA, pp 177–179. https://doi.org/10.1109/TPS-ISA58951.2023.00030
157. Sarker MK, Zhou L, Eberhart A, Hitzler P (2022) Neuro-symbolic artificial intelligence: current trends. AI Commun 34(3):197–209. https://doi.org/10.3233/AIC-210084
158. Bulla C, Birje MN (2021) Improved data-driven root cause analysis in fog computing environment. J Reliab Intell Environ 8:359–377
159. Balla J, Huang S, Dugan O et al (2023) AI-assisted discovery of quantitative and formal models in social science. arXiv preprint arXiv:2210.00563
160. Gangopadhyay B, Soora H, Dasgupta P (2022) Hierarchical program-triggered reinforcement learning agents for automated driving. IEEE T-ITS 23(8):10902–10911. https://doi.org/10.1109/TITS.2021.3096998
161. Li Q, Zhu Y, Liang Y et al (2024) Neural-symbolic recursive machine for systematic generalization. arXiv preprint arXiv:2210.01603
162. Kumar SAP, Bao S, Singh V, Hallstrom J (2019) Flooding disaster resilience information framework for smart and connected communities. J Reliab Intell Environ 5:3–15. https://doi.org/10.1007/s40860-019-00073-2
163. Harmon I, Marconi S, Weinstein B et al (2022) Injecting domain knowledge into deep neural networks for tree crown delineation. TGRS 60:1–19. https://doi.org/10.1109/TGRS.2022.3216622
164. Santhalingam PS, Hosain AA, Zhang D et al (2020) mmASL: environment-independent asl gesture recognition using 60 ghz millimeter-wave signals. IMWUT. https://doi.org/10.1145/3381010
165. Gruyer D, Magnier V, Hamdi K et al (2017) Perception, information processing and modeling: critical stages for autonomous driving applications. Annu Rev Control 44:323–341. https://doi.org/10.1016/j.arcontrol.2017.09.012
166. Dobosz K, Duch W (2010) Understanding neurodynamical systems via fuzzy symbolic dynamics. Neural Netw 23(4):487–496. https://doi.org/10.1016/j.neunet.2009.12.005. (**ICANN 2008**)
167. Lu HL, Ong K, Chia P (2000) An automated ecg classification system based on a neuro-fuzzy system. In: Computers in cardiology 2000, vol 27 (Cat. 00CH37163), pp 387–390. https://doi.org/10.1109/CIC.2000.898538
168. Han L, Srivastava MB (2024) An empirical evaluation of neural and neuro-symbolic approaches to real-time multimodal complex event detection. arXiv preprint arXiv:2402.11403
169. Chen X, Chen G, Ge L, Huang B et al (2021) Global oceanic eddy identification: a deep learning method from argo profiles and altimetry data. Front Mar Sci 8:646926
170. Wilson A, Kumar A, Jha A, Cenkeramaddi LR (2021) Embedded sensors, communication technologies, computing platforms and machine learning for uavs: a review. IEEE Sens J 22(3):1807–1826
171. Cho A, Kang Y-S, Park B-J, Yoo C-S, Koo S-O (2011) Altitude integration of radar altimeter and GPS/INS for automatic takeoff and landing of a UAV. In: ICCAS, pp 1429–1432
172. Hoang ML, Carratù M, Paciello V, Pietrosanto A (2021) Body temperature-indoor condition monitor and activity recognition by mems accelerometer based on iot-alert system for people in quarantine due to COVID-19. Sensors 21(7):2313
173. Ahad MAR, Antar AD, Ahmed M (2020) IoT sensor-based activity recognition. IEEE Trans Syst Man Cybern Part C (Appl Rev) 2:790–808

174. Moin A, Aadil F, Ali Z, Kang D (2023) Emotion recognition framework using multiple modalities for an effective human-computer interaction. J Supercomput 79(8):9320–9349
175. Wan Z, Liu C-K, Yang H, et al. (2024) Towards cognitive AI systems: a survey and prospective on neuro-symbolic AI. arXiv:2401.01040
176. Serafini L, Garcez A (2016) Logic tensor networks: deep learning and logical reasoning from data and knowledge. arXiv:1606.04422
177. Kim J, Canny J (2017) Interpretable learning for self-driving cars by visualizing causal attention
178. Carlini N, Mishra P, Vaidya T, *et al.* (2016) Hidden voice commands. In: 25th USENIX security symposium, pp 513–530
179. Shokri R, Stronati M, Song C, Shmatikov V (2017) Membership inference attacks against machine learning models. In: 2017 IEEE symposium on security and privacy (SP), pp 3–18
180. Song Y, Kim T, Nowozin S et al (2017) Pixeldefend: leveraging generative models to understand and defend against adversarial examples. arXiv:1710.10766
181. Michel-Delétie C, Sarker MK (2024) Neuro-symbolic methods for trustworthy AI: a systematic review. Neurosymb Artif Intell 0(1):1–14
182. Agiollo A, Omicini A (2023) Measuring trustworthiness in neuro-symbolic integration. In: FedCSIS, pp 1–10. https://doi.org/10.15439/2023F6019
183. Sheth A, Roy K (2023) Neurosymbolic value-inspired AI (why, what, and how). arXiv preprint arXiv:2312.09928
184. Guan J (2019) Artificial intelligence in healthcare and medicine: promises, ethical challenges and governance. Chin Med Sci J 34(2):76–83
185. Kleinberg J, Mullainathan S, Raghavan M (2016) Inherent trade-offs in the fair determination of risk scores. arXiv:1609.05807
186. Cunnington D, Law M, Russo A et al (2021) Towards neural-symbolic learning to support human-agent operations. In: FUSION, pp 1–8