Cyber-Physical AI: Systematic Research Domain for Integrating AI and Cyber-Physical Systems

SANGHOON LEE, JIYEONG CHAE, HAEWON JEON, TAEHYUN KIM, YEONG-GI HONG, DOO-SIK UM, TAEWOO KIM, and KYUNG-JOON PARK, Electrical Engineering and Computer Science, Daegu Gyeongbuk Institute of Science and Technology, Daegu, Republic of Korea

The integration of Cyber-Physical Systems (CPS) and AI presents both opportunities and challenges. AI operates on the principle that "good things happen probabilistically," while CPS adheres to the principle that "all bad things must not happen," requiring uncertainty-awareness. Furthermore, the difference between AI's resource accessibility assumption and CPS's resource limitations highlights the need for resource-awareness. We introduce Cyber-Physical AI (CPAI), an interdisciplinary sub-field of AI and CPS research, to address these constraints. To the best of our knowledge, CPAI is the first research domain on CPS-AI integration. We propose a 3D classification schema of CPAI: Constraint (C), Purpose (P), and Approach (A). We also systematize the CPS-AI integration process into three phases and nine steps. By analyzing 104 studies, we highlight nine key challenges and insights from a CPAI perspective. CPAI aims to unify fragmented studies and provide guidance for reliable and resource-efficient integration of AI as a component of CPS.

CCS Concepts: • **General and reference** → **Surveys and overviews**; • **Computer systems organization** → **Embedded and cyber-physical systems**; • **Computing methodologies** → **Artificial intelligence**; **Machine learning**;

Additional Key Words and Phrases: Artificial Intelligence, Cyber-Physical Systems, Machine Learning, Embedded Systems, System Integration, Uncertainty Adaptive, Resource-Awareness, Real-time, Automation, Augmentation

This work was supported by Korea Research Institute for defense Technology planning and advancement (KRIT) grant funded by the Korea government (DAPA (Defense Acquisition Program Administration)) (KRIT-CT-22-040, Heterogeneous Satellite constellation based ISR Research Center, 2022) and the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (2023R1A2C2003901).

Authors' Contact Information: Sanghoon Lee, Electrical Engineering and Computer Science, Daegu Gyeongbuk Institute of Science and Technology, Daegu, Republic of Korea; e-mail: leesh2913@dgist.ac.kr; Jiyeong Chae, Electrical Engineering and Computer Science, Daegu Gyeongbuk Institute of Science and Technology, Daegu, Republic of Korea; e-mail: cowldud3@dgist.ac.kr; Haewon Jeon, Electrical Engineering and Computer Science, Daegu Gyeongbuk Institute of Science and Technology, Daegu, Republic of Korea; e-mail: wmjhw0624@dgist.ac.kr; Taehyun Kim, Electrical Engineering and Computer Science, Daegu Gyeongbuk Institute of Science and Technology, Daegu, Republic of Korea; e-mail: 3nz85ag@dgist.ac.kr; Yeong-Gi Hong, Electrical Engineering and Computer Science, Daegu Gyeongbuk Institute of Science and Technology, Daegu, Republic of Korea; e-mail: hykfun@dgist.ac.kr; Doo-Sik Um, Electrical Engineering and Computer Science, Daegu Gyeongbuk Institute of Science and Technology, Daegu, Republic of Korea; e-mail: uds0909@dgist.ac.kr; Taewoo Kim, Electrical Engineering and Computer Science, Daegu Gyeongbuk Institute of Science and Technology, Daegu, Republic of Korea; e-mail: vlzkcb0206@dgist.ac.kr; Kyung-Joon Park (corresponding author), Electrical Engineering and Computer Science, Daegu Gyeongbuk Institute of Science and Technology, Daegu, Republic of Korea; e-mail: kjp@dgist.ac.kr.

ACM 2378-9638/2025/4-ART19
https://doi.org/10.1145/3721437

ACM Reference format:
Sanghoon Lee, Jiyeong Chae, Haewon Jeon, Taehyun Kim, Yeong-Gi Hong, Doo-Sik Um, Taewoo Kim, and Kyung-Joon Park. 2025. Cyber-Physical AI: Systematic Research Domain for Integrating AI and Cyber-Physical Systems. *ACM Trans. Cyber-Phys. Syst.* 9, 2, Article 19 (April 2025), 33 pages.
https://doi.org/10.1145/3721437

1 Introduction

The advancement of **Internet of Things (IoT)** technology and the development of communication and computing technologies have widely utilized **Cyber-Physical Systems (CPS)**, ranging from personal devices to large-scale engineering systems. CPS represents a paradigm shift rather than merely a technology application. CPS refers to systems where cyber systems, represented by computing resources, interact with physical and engineering systems through a network to perform monitoring, control, and adjustment tasks [1]. Examples of CPS include smart factories, autonomous robots, smart cities, and more.

The increasing connectivity and complexity within CPS pose challenges to traditional operations, while the significant amount of data generated by CPS provides opportunities for the application of AI [2, 3]. This leads to efforts to integrate AI as a component of CPS, aiming for operational methods that surpass traditional approaches. However, CPS inherently requires adaptability to uncertainty and resource-awareness.

AI is fundamentally based on statistics, operating under the premise that "good things happen probabilistically," while CPS evolves from a domain that is theoretically well-defined, adhering to the principle that "all bad things must not happen" [4]. In the domain of AI, systems like large language models, such as ChatGPT, illustrate AI's probabilistic nature by occasionally generating inaccurate outputs, with users advised to verify crucial information due to the expected margin of error [5]. This poses a major challenge of AI-CPS integration, where much more stringent requirements must be met, such as autonomous robots and smart factories [6]. In such CPS contexts, even minor errors can have severe consequences, leading to system failure or physical harm. Especially due to the complexity of interactions among components that make CPS prone to accidents [7], the integration with AI can encounter increased resistance.

Within the domain of AI, it's commonly assumed that "resources are always accessible and intended for AI" [8]. On the other hand, in the CPS principle, "resources are a shared commodity, therefore inherently limited" [9, 10]. As the complexity of components and processes in CPS increases, resource-efficient design in CPS is becoming increasingly important [11]. However, the mainstream of AI research is biased toward achieving higher accuracy rather than considering resource efficiency [12], and consideration of resources is also focused primarily on the performance of AI (mainly, training speed) [13]. In other words, mainstream AI research does not assume that AI has to share resources with other components as one of the components of CPS nor does it consider that there may be tasks more critical than the AI task. This oversight presents significant challenges for CPS-AI integration.

The majority of research on CPS-AI integration focuses on adapting AI to fit the constraints of CPS or modifying CPS to accommodate AI. While there has been extensive research on the CPS-AI integration, most survey research on this integration has focused on investigating specific technical fields [14–16] or particular application areas [17, 18]. They mainly categorize and discuss specific AI technologies used, but do not address the constraints of AI integration. To the best of our knowledge, there has not yet been a comprehensive survey or review that focuses specifically on the CPS-AI integration process. In order to promote the CPS-AI integration, it is critical to establish a systematic research domain that focuses on integration itself. In this context, we introduce a novel concept

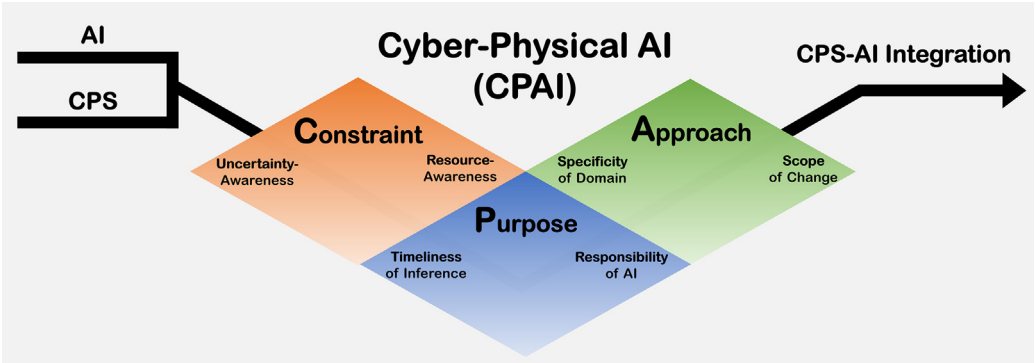

Fig. 1. Overview of CPAI.

called **Cyber-Physical AI (CPAI)**, which encompasses research that arises during the CPS-AI integration process, focusing on the constraints. Our article makes the following contributions:

—We clearly define the research domain, CPAI, and present a classification schema.
—We systematically outline the entire CPS-AI integration process.
—We analyze existing studies and summarize nine key challenges.
—We categorize 104 studies and provide unique insights from a CPAI perspective.

The rest of this article is organized as follows: Section 2 clearly describes the definition and scope of CPAI, presents the classification schema of CPAI research. Section 3 systematizes the entire process of CPS-AI integration, dividing it into three phases, each consisting of three detailed steps. Section 4 introduces the key challenges that arise during the CPS-AI integration process and categorizes studies from the CPAI perspective. Section 5 concludes the article.

2 CPAI

In Section 2, we introduce a new concept, CPAI. Furthermore, we present "Three Dimensions of CPAI," a framework for systematically categorizing CPAI research.

2.1 Definition of CPAI

CPAI is an interdisciplinary research domain that considers the challenges that arise during CPS-AI integration. CPAI represents a sub-field within AI research, encompassing various technologies and methodologies studied for the purpose of integrating AI into CPS. Figure 1 illustrates an overview of CPAI. In the CPS-AI integration, the primary focus of CPAI is not on the performance achieved through the integration, but on addressing the challenges that arise during the integration process. Thus, CPAI research must involve the modification of AI technologies or other CPS components to address specific constraints. We emphasize that simply applying AI to CPS without any modifications to CPS and AI does not fall within the scope of CPAI.

2.2 Three Dimensions of CPAI

The global problem of CPAI is to address the challenges arising from the CPS-AI integration. Consequently, CPAI can be broadly categorized into three fundamental sub-problems: What causes these challenges? Why must they be resolved? How can they be overcome? In this context, we introduce "Three Dimensions of CPAI" as a schema for categorizing CPAI research. As illustrated

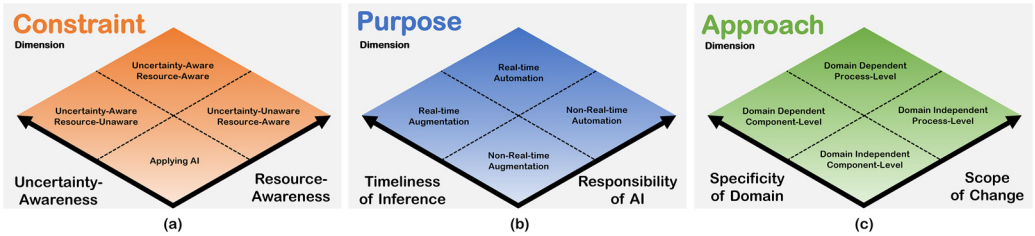

Fig. 2. Three dimensions of CPAI.

in Figure 2, CPAI research encompasses three distinct aspects, reflecting the acronym: Constraint (C), Purpose (P), and Approach (A).

The Constraint dimension represents the underlying constraints that cause challenges in CPS-AI integration. The Purpose dimension represents the purpose of the AI component in the CPS-AI integration. Constraints can be strengthened or weakened based on the purpose of the AI component. The Approach dimension emphasizes methodologies on how to overcome the challenges of integration under these constraints and purposes. Detailed explanations for each dimension can be found in Sections 2.3–2.5.

2.3 Constraint Dimension

CPAI research aims to address the challenges that arise during CPS-AI integration. These challenges are inherent to the constraints of CPS and AI. Thus, constraints serve as a motivation for pursuing CPAI research. If a study does not consider any constraints, it is merely applying AI and does not fall under the CPAI domain. While taxonomies for constraint aspects may include ethics, scalability, and economics, these factors are fundamentally governed by uncertainty and resource constraints. Therefore, we categorize the constraints in CPAI research into two main categories: Uncertainty and Resource.

2.3.1 Uncertainty. AI and CPS have been studied from fundamentally different perspectives. In CPS, most processes demand deterministic behavior within strict boundaries. Conversely, AI has evolved from statistical principles and inherently embraces statistical indeterminacy [4]. This fundamental difference is one of the key constraints that make CPS-AI integration a complex challenge. Additionally, CPS environments possess inherent uncertainties due to their physical systems and networks. In this context, we define uncertainty-aware research.

— *Uncertainty-awareness* refers to the consideration of uncertainties, broadly categorized into four types: model, data, network, and physical. Model uncertainty includes inherent uncertainty in the AI model due to stochastic processes and limitations in algorithmic design. Data uncertainty pertains to the reliability of AI, which heavily depends on the quality and quantity of the training data. Network uncertainty involves variability and unreliability in network performance, such as latency and bandwidth fluctuations. Physical uncertainty encompasses variations and unpredictability in the physical components and environment of the CPS.
— *Uncertainty-unawareness* refers to the lack of consideration for uncertainty constraints. These studies often assume sufficient data, fixed conditions, and predictable environments.

2.3.2 Resource. Mainstream AI research focuses on achieving higher accuracy and faster training speeds, without assuming AI as one component of a larger system that shares resources. Consequently, the resource demands of AI are exponentially increasing, often overshadowing considerations of resource efficiency [8]. For successful CPS-AI integration, it's essential to clearly

recognize the substantial resource consumption by AI when integrated into CPS. In this context, we define resource-aware research.

— *Resource-awareness* refers to the consideration of resources, broadly divided into four types: computing, network, sensor, and human. Network resources are required to send and receive data, ensuring data transfer efficiency. Computing resources are needed for AI training and inference, impacting processing capability and speed. Sensor resources installed in the CPS are crucial for data acquisition. Human resources involve experts and operators, essential for the CPS-AI integration process.
— *Resource-unawareness* refers to the lack of consideration of resource constraints. These studies often assume high-performance networks, computing environments or sufficient physical resources.

2.4 Purpose Dimension

Even within the same domain, the impact and extent of constraints can vary depending on the purpose of the AI component.

The purpose dimension can be classified based on factors such as the application domain (e.g., healthcare, manufacturing, energy) or the values being pursued (e.g., efficiency, safety, sustainability). However, the timeliness of inference and the level of AI's responsibility most directly influence the relevant constraints. For example, AI applications requiring real-time inference may impose strict resource constraints, while automation tasks such as control or allocation demand significant attention to uncertainties. Therefore, we classify the purpose in CPAI research into two main categories: Timeliness and Responsibility.

2.4.1 Timeliness of Inference. AI predominantly encompasses two phases: training and inference. Depending on the purpose of application, the temporal requirements for inference differ significantly. Such requirements impose stricter resource constraints to the system. We categorize CPAI research into two categories based on the temporal requirements of AI:

— *Real-time* refers to AI applications where inference requires real-time processing. This involves dynamically updating models with new data and making immediate inferences through real-time data. It can demand higher levels of resource constraints.
— *Non-real-time* refers to AI inference without temporal requirements. Studies with no mention of temporal requirements are categorized as non-real-time. In this application, resource constraints can be comparatively less severe.

2.4.2 Responsibility of AI. The responsibility of the AI component within the CPS is a critical factor. AI can either make direct decisions, or it provides information indirectly. Automating the CPS with direct AI decision-making increases the responsibility of AI within the CPS. We categorize CPAI research into two categories based on the responsibility of AI:

— *Automation* refers to AI making independent decisions within CPS and executing them to directly control or operate the system. Such applications can significantly enhance the system's efficiency and autonomy. However, relying on AI's decisions increases the risk associated with uncertainties.
— *Augmentation* refers to AI supporting the function of CPS with the information it provides. This includes detection and prediction tasks. Such applications can reduce the risks associated with uncertainty constraints, but its efficiency is significantly influenced by the existing components.

2.5 Approach Dimension

CPAI research proposes new approaches to address the constraints that arise from CPS-AI integration used for specific purposes. The approaches correspond directly to the methodologies of the studies. These approaches can be categorized based on the algorithms used (e.g., machine learning, optimization theory) and the methods of verification (e.g., mathematical proofs, simulations, real-world testing). However, various algorithms and verification techniques are now frequently combined. To ensure clarity in classifying the methodologies of each study, we classify the approaches of CPAI research into two main categories: Specificity and Scope.

2.5.1 Specificity of Domain. CPS is applied across various domains, including smart factories, cities, agriculture, healthcare, and military. Therefore, CPAI research sometimes utilizes domain knowledge or is inspired by the characteristics of the domain. The domain specificity of CPAI research assists in evaluating the generality of the study. We categorize CPAI research into two categories based on the domain specificity:

— *Domain-dependent* approach refers to studies whose main ideas are specific to a particular domain. These studies are developed through a deep understanding of the unique characteristics and problems of that domain. They can offer more precise and efficient solutions, but this often results in a limited scope of application.
— *Domain-independent* approach refers to studies in which the main ideas are independent of the target domain. These studies are developed using domain-agnostic solutions such as additional AI methods. While they have high applicability, they may have limitations in addressing the unique requirements of specific domains.

2.5.2 Scope of Change. CPS involves the interaction of diverse components across cyber systems, physical systems, and networks. To address constraints, approaches can either modify specific components or transform processes, which refers to the interaction of multiple components. We categorize CPAI research into two categories based on the scope of change the research brings to CPS:

— *Process-level* approach refers to studies that aim for fundamental transformations in the processes of CPS. These studies include altering existing processes or introducing new ones. These approaches can be difficult to apply, as they sometimes require architectural changes or additional resources.
— *Component-level* approach refers to studies that focus on modifying a single component within CPS. Such studies do not necessitate changes to other components or processes. They can be applied more easily within existing CPS systems, but their potential for improvement is limited.

3 Process of CPS-AI Integration

In Section 3, we systematize the entire process of CPS-AI integration. Figure 3 provides an overview of the integration process. We represent the integration process using the waterfall model, commonly used in traditional software development [19]. We categorize the process into three distinct phases: design, development, and deployment. Each phase consists of three detailed steps.

3.1 Design Phase

The design phase is the initial process in the CPS-AI Integration. This phase includes planning for the integration, acquiring, and processing data necessary for the development of AI. Projects involving AI-CPS integration often fail due to deficiencies in the design phase, such as data quality

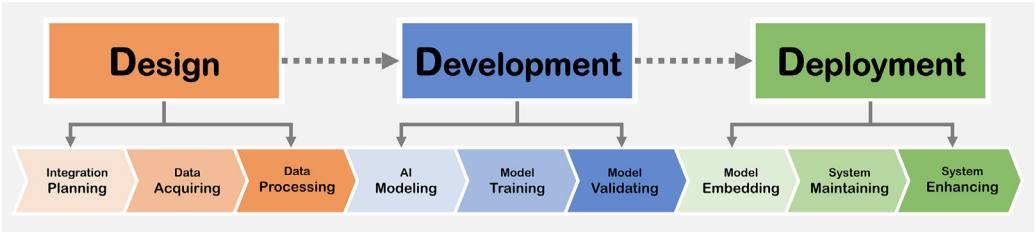

Fig. 3. Process of CPS-AI integration.

issues [20] and the cost of data processing. To preempt integration failures, it is crucial to clearly understand the constraints of the integration during the design process.

3.1.1 Planning. The planning step is about understanding the necessity and feasibility of CPS-AI integration and establishing a clear plan for the subsequent eight steps. These plans can progress in a forward direction, from data acquisition to continuous improvement, or in reverse. A crucial element in the planning step is the clear recognition of the target system's constraints and the integration's forms. Depending on the forms of the integration, the requirements for integration may vary. Similarly, the system's constraints dictate the considerations that must be addressed.

3.1.2 Acquiring. The acquiring step involves collecting datasets essential for AI training. A pivotal aspect of this step is striking a balance between the data quality demanded by the AI and the cost of data collection from the CPS. The AI's requirements dictate the minimum data quality necessary, while the constraints of the target system significantly influence the cost of CPS data acquisition. In general, the acquisition of higher-quality data mitigates the uncertainty constraints of the AI but necessitates greater expenditure.

3.1.3 Processing. The processing step refers to the process of transforming the data collected during the acquisition step into meaningful features and formats suitable for AI. The data collected from the CPS can be unbalanced, noisy, high-dimensional, heterogeneous, and sometimes unlabeled. Physical uncertainties in the CPS can amplify data uncertainty, making it essential to conduct data processing that takes into account these physical uncertainties.

3.2 Development Phase

The development phase is the intermediate process in CPS-AI integration. In this phase, the appropriate AI methods, training, and validation are selected based on the plans and data prepared during the design phase. The development phase has been a primary focus of traditional AI research. However, most research on the development phase has overlooked the constraints inherent to the design and deployment phases. It is imperative that CPAI research on the development phase should be integrated with the constraints related to the other phases.

3.2.1 Modeling. The modeling step involves designing an AI model that takes into account system specifications and processed data. A crucial element of this step is the acknowledgment that "highest performance does not necessarily equate to the most suitable choice." Even if an AI model consistently demonstrates superior performance, it may not be conducive to integration if it lacks robustness against uncertainties or demands excessive resources. It is therefore imperative to undertake AI modeling from the perspective of CPAI.

3.2.2 Training. The training step refers to the process of training the AI model designed during the modeling step. Depending on the purpose of integration, training can be conducted using offline

learning, which utilizes a fixed dataset, or online learning, where data are generated in real-time. During the training step, it is critical to consider the impact of CPS uncertainties and the resource demands of training.

3.2.3 Validating. The validating step involves evaluating the trained AI model using a validation dataset. Traditional AI research has focused on validation in terms of classification or prediction accuracy, focusing on average inference performance. However, this approach often overlooks critical issues such as reliability and real-time performance. In the context of CPAI, validation should assess robustness to environmental changes or adversarial attacks, along with response time and resource utilization.

3.3 Deployment Phase

The deployment phase presents the final process in CPS-AI integration. During this phase, the AI model is embedded, maintained, and enhanced within the CPS. This phase is influenced by practical challenges rather than theoretical limitations. Research on the deployment phase has been relatively neglected [21] as integrating AI into real systems and maintaining them long-term pose significant challenges at the laboratory level. However, it is crucial for successful CPS-AI integration.

3.3.1 Embedding. The embedding step involves integrating the AI into the actual components of the CPS. Successful embedding means that the AI's capabilities, from data collection to inference, align with the integration goals. In addition, the embedding should minimize resource consumption and mitigate any negative impact on existing CPS processes. These considerations are more critical for real-time applications and may require process changes in the CPS.

3.3.2 Maintaining. The maintenance step refers to preserving the functionality of the AI integrated into the CPS and ensuring its continued performance. Maintaining AI—not using AI for maintenance—is critical to the commercial success of AI applications [21]. Maintaining AI fundamentally requires addressing the complexity and uncertainty of AI systems, which are often considered "black boxes," and it must be feasible from a resource perspective.

3.3.3 Enhancing. The enhancing step involves modifying or creating AI systems based on environmental changes or additional requirements. While maintaining addresses temporary changes, enhancing focuses on adapting to permanent changes. The enhancing step allows AI systems to respond quickly to environmental changes and is more resource-efficient than a completely new CPS-AI integration process.

4 Challenges of CPAI

In Section 4, we introduce the nine major CPAI challenges that act as obstacles to CPS-AI integration. Figure 4 depicts the CPAI challenges encountered during CPS-AI integration. In the design phase, where data are collected and processed in the CPS, AI faces challenges such as data imbalance, data scarcity, and insufficient labels. In the development phase, where AI is trained and validated, it must be prepared for issues like drift, data loss, and unreliable inference. Finally, in the deployment phase, where the developed AI is embedded and operated in the CPS, compute and network limitations and adversarial attacks are important considerations.

As previously mentioned, CPAI is an interdisciplinary field, bridging CPS and AI. While these challenges emerge in other domains that apply AI, their solutions may not directly translate to CPS. In CPS environments, the same challenges can become more pronounced due to unique constraints. In this section, we highlight approaches to address these challenges, and examine the relationships

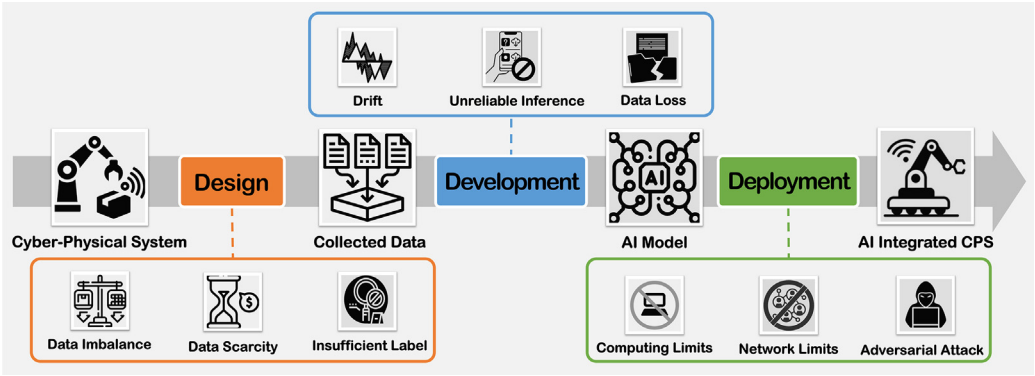

Fig. 4. Challenges of CPAI in CPS-AI integration process.

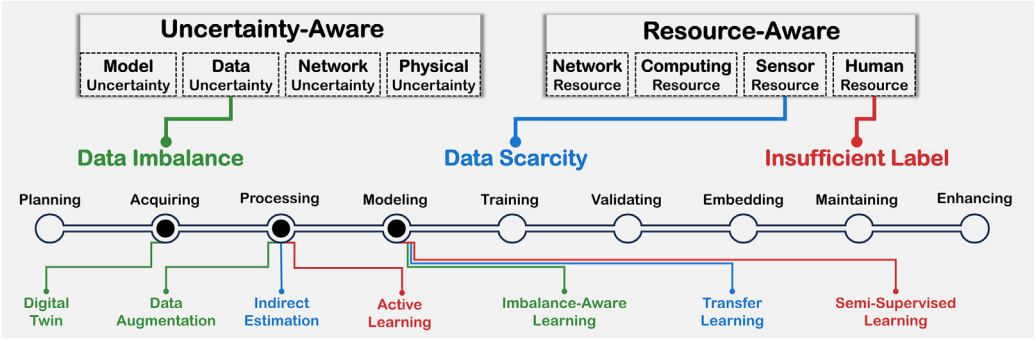

Fig. 5. Challenges and solutions in design phase.

among constraints, purposes, and approaches. To achieve this, we review 104 studies focusing on these CPAI challenges and map them to the three dimensions of CPAI.

Figure 5 summarizes the challenges and solutions in the design phase from the perspective of constraints and CPS-AI integration. To address the issue of data imbalance, data can be collected through a **Digital Twin (DT)** that imitates the physical systems of CPS. Furthermore, the data augmentation process may be utilized to generate new data from the collected data, thereby achieving balance. Another solution is the utilization of AI models that are specifically designed to handle imbalanced data. To address the issue of data scarcity, it is possible to estimate the unavailable data from existing indirect data or to utilize transfer learning models that have been developed with data from other domains. In the event of insufficient labels, potential solutions include the application of **Active Learning (AL)** during the processing step to perform minimal labeling or using semi-supervised learning models that combine unsupervised and supervised learning. Details of the challenges and solutions in the design phase can be found in Sections 4.1–4.3.

4.1 Data Imbalance

As is the case with CPS-AI integration, data play a pivotal role in other domains as well. This is due to the fact that the performance of AI models is dependent upon the quality and quantity of the training data. Successful integration requires large volumes of data with consistent distribution. However, certain events such as failures, intrusions, and malfunctions in physical systems are much

rarer compared to normal conditions. As a result, the data collected in CPS can exhibit extreme class distributions. This extreme "data imbalance" between classes can make AI training impossible or lead to model overfitting.

4.1.1 DT. One method to address the data imbalance problem is the use of DT. A DT is a virtual and computerized counterpart within a cyber system that corresponds to a physical system [22]. While DTs are primarily used for real-time monitoring of sensor data and synchronization of system changes, they can also be used to generate data for AI training. There are significant costs and risks associated with inducing attacks or failures in an actual physical system, while the costs and risks are significantly lower in a virtual DT environment. By simulating abnormal situations that could affect the physical system, DTs can help collect data on these scenarios, thereby addressing the data imbalance problem.

Castellani et al. [23] suggest employing a DT to generate normal and abnormal operational data to mitigate the issue of rare data samples being misclassified as anomalous in unsupervised machine anomaly detection. Liebert et al. [24] and Markovic et al. [25] propose a technique to generate abnormal situation data through a DT of a modular cyber-physical production system by independently modeling and combining each system component. Xu et al. [26] increase the volume of anomalous data in train control using knowledge distillation based on autoencoders and DT. Galvan et al. [27] use a DT integrating the Gazebo simulator, QGround Control, and PX4 Autopilot software to collect abnormal data for **Unmanned Aerial Vehicle (UAV)** anomaly detection.

4.1.2 Data Augmentation. While DT addresses the data imbalance problem during the acquiring step, data augmentation serves as a solution during the processing step. Data augmentation involves generating additional training data by applying transformations to the original samples while preserving their labels [28]. Numerous studies on data augmentation have been conducted in the computer vision domain, focusing on image and video data [29]. In CPS, not only image and video data are generated but also physical sensor data such as sound, temperature, velocity, and pressure, and operational data such as network and control information. As a result, existing data augmentation methods may not always be applicable, creating opportunities for the development of augmentation techniques tailored to specific domains.

A **Generative Adversarial Network (GAN)** is a technique that generates realistic new data through the competition between two neural networks: the generator, which generates fake data, and the discriminator, which distinguishes between the generated data and the real data. This method is widely used for data augmentation. In [30], GAN-based data augmentation is used to detect defects in images of metal manufacturing parts. In [31], GAN-based data augmentation is used to detect water leaks in water pipes. In [32], GAN-based data augmentation is applied to defect data consisting of 590 sensor measurements from a semiconductor wafer production line. Sometimes, domain-specific augmentation methods are required. In [33], audio-specific data augmentation methods such as pitch shift, white noise injection, and frequency masking are applied to sound data related to machine defects. Vo et al. [34] propose data augmentation techniques specifically designed for UAV network traffic flow. The data are divided into sets that are easy or difficult to classify based on labels, with the easy set compressed and the difficult set expanded to generate new data.

4.1.3 Imbalance-Aware Learning. In the modeling step, the solution involves accepting data imbalance and designing AI models tailored to the data. A common example of imbalance-aware learning is **One-class Classification (OC)**. OC is used in scenarios where the training data contains a single dominant class, with other classes being sparse or poorly defined. In [35], a **Deep Neural Network (DNN)-Long Short-Term Memory (LSTM)**-based OC model is proposed to detect

errors in the time series data of temperature and pressure from the die casting process. Yang et al. [36] propose a broad learning system-based OC classifier to detect intrusions in complex network traffic data. Another solution is to use expert domain knowledge. Ye et al. [37] propose a weighted random forest that provides stable predictions despite data imbalance by extracting 16 technical parameters and 7 electrical parameters that affect defects in the continuous casting process based on expert domain knowledge.

4.2 Data Scarcity

When integrating CPS-AI, data scarcity may arise when the training data is insufficient. This may result from high costs, physical constraints of sensor installation, or insufficient time for sensing. Data scarcity describes the lack of sufficient data across all classes, which can make AI model training infeasible. To address this challenge, similar to dealing with data imbalance, DT methods can be employed. In [38], image data are generated by a DT consisting of 3D models created by CAD software, virtual cameras, and virtual robots. In [39], a DT is constructed by reconstructing the system model using the **Remaining Useful Life (RUL)** of the machine and simulating physical sensor data for CNC machines.

4.2.1 Transfer Learning. A common solution to data scarcity is AI modeling based on transfer learning. In transfer learning, the parameters and structure of a model trained on existing domain data are used as initial values for learning in a new domain. By freezing or fine-tuning some layers of the existing model to adapt to the new domain, an efficient model can be achieved with limited data. In CPS, data scarcity is common, making the application of transfer learning a widely researched topic.

An example of a transfer learning application is real-time production progress prediction in manufacturing systems. For current orders, there is often no historical data available, leading to data scarcity problems. Therefore, transfer learning strategies that extract domain features from the production data of other orders and adapt to the specific characteristics of the data of the current order are used. In [40], a deep auto encoder is used to extract domain features and a deep belief network is used to design a predictor. Liu et al. [41] use **Convolution Neural Networks (CNN)** to extract domain features and proposes a predictor combined with LSTM.

In CPS, heterogeneous systems sometimes require common applications. In such cases, data collected from a single system may not be sufficient and transfer learning can be used to solve this problem. In [42], CNN-based transfer learning with non-manufacturing data is used to build a fault detection system for rolling bearings in environments where it is difficult to obtain consistent data for each machine due to process characteristics. Wang et al. [43] propose a ResNet-based transfer learning strategy to overcome the lack of attack samples in the design of a network attack detector for CPS, using open data with a large number of attack samples. Conversely, there are cases where different applications are required within the same system. In [44], a federated transfer learning framework is introduced to effectively solve the data scarcity problem in smart manufacturing applications. When a new application is created, a similar model from existing applications is selected at the central server to perform transfer learning.

4.2.2 Indirect Estimation. Another interesting strategy is to use indirect data, such as data from neighboring machines or operational data, when it is not possible to collect data directly from a machine. Putnik et al. [45] propose a strategy for learning the state of a machine using indirect data. This includes metrics such as production uptime, batch size, cycle count, shift change frequency, setup count, and time lost. This approach is used in scenarios where the infrastructure to collect direct state data (e.g., vibration and temperature) is not available. Kim et al. [46] present a data-driven strategy for estimating mean time to repair and mean time between failures using data

from adjacent machines. This strategy is used when direct data collection from a specific machine is not feasible.

4.3 Insufficient Label

Many AI approaches require labeled datasets. In particular, commonly used deep learning methods require a large number of labeled samples. In CPS, labeling can sometimes be automated, but in most cases, it requires manual labeling by experts. Therefore, despite having sufficient data, there is often a shortage of labeled data due to limited human resources. This labeling shortage problem can be addressed by transfer learning, as mentioned above. To address the labeling shortage in rolling bearing fault detection, [47] propose a processing technique to generate auxiliary samples for the target domain using LSTM on data from the source domain, which has abundant labeling. Qin et al. [48] introduce a new transfer learning model for fault detection in industrial data, called the parameter sharing adversarial domain adaptation network. This model combines the fault classifier and the domain classifier through parameter sharing, which reduces the complexity of the network and minimizes the training cost.

4.3.1 AL. AL is a method to address the need for large amounts of labeled data by intelligently querying labels during learning, thereby reducing the need for labeled data [49]. AL is applicable when there is an opportunity to engage in the collection and selection of data, following a strategy that repeatedly selects the most informative samples for labeling. **Bayesian Active Learning by Disagreement (BALD)** is a prominent AL method that selects data points to label based on the maximum expected reduction in model uncertainty by exploiting disagreement among multiple model predictions. Shim et al. [50] employ an AL strategy using BALD for wafer map anomaly detection in semiconductor manufacturing processes. Todic et al. [51] propose a hybrid BatchBALD strategy for energy analysis in power systems, combining a pool-based strategy (targeting large static samples) with a stream-based strategy (targeting real-time transmitted samples). There are other AL approaches as well. For example, [52] introduce an active anomaly detection strategy in printed circuit board manufacturing environments that propagates sample representativeness through a graph structure while empirically combining uncertainty and representativeness strategies.

4.3.2 Semi-Supervised Learning. Methods that use both labeled and unlabeled data are classified as semi-supervised learning. For more detailed studies on semi-supervised learning, see [53]. Cyber attack detection is a typical example of labeling constraints in CPS. Huda et al. [54] propose a semi-supervised approach that combines unsupervised clustering to extract information about unknown cyber attacks and integrates this with supervised learning classifiers such as support vector machine and random forest. This method improves attack detection accuracy without requiring expert effort to update the detection engine's database. Wang et al. [55] introduce a deep semi-supervised learning framework for CPS to identify false fault data. It uses distance-based clustering for labeled data and an isolation forest for unlabeled data, incorporating a pseudo-labeling strategy. Pseudo-labeling uses predicted labels for unlabeled data to proceed with learning.

Table 1 maps the studies presented in Sections 4.1–4.3 to the three dimensions of CPAI.

The three main challenges arising during the design phase have all been studied in the context of AI augmentation. This is due to the significant risks posed by implementing AI-based automation in environments with insufficient data and labeling.

Among the approaches to addressing data imbalance, data augmentation is the most accessible. Augmentation requires changes to only certain components of data processing and offers a variety of domain-independent techniques. However, training imbalance-aware models is less accessible as it is often limited to specific situations (e.g., where only OC classification is required) or relies on

Table 1. Studies for Challenges of Design Phase

Challenge	Method	Ref.	Constraint		Approach		Purpose	
			Uncertainty-Awareness	Resource-Awareness	Scope of Change	Specificity of Domain	Timeliness	Responsibility
Data Imbalance (4.1)	DT	[23]	✓		Process	Dependent	Real-time	Augmentation
		[24]	✓		Process	Dependent	Non-real	Augmentation
		[25]	✓		Process	Dependent	Non-real	Augmentation
		[26]	✓		Process	Independent	Real-time	Augmentation
		[27]	✓		Process	Dependent	Real-time	Augmentation
	Data Augmentation	[30]	✓		Component	Independent	Non-real	Augmentation
		[31]	✓	✓	Component	Independent	Non-real	Augmentation
		[32]	✓		Component	Independent	Non-real	Augmentation
		[33]	✓		Component	Dependent	Non-real	Augmentation
		[34]	✓	✓	Component	Independent	Real-time	Augmentation
	Imbalance-Aware Learning	[35]	✓		Component	Independent	Non-real	Augmentation
		[36]	✓	✓	Component	Independent	Real-time	Augmentation
		[37]	✓		Component	Dependent	Real-time	Augmentation
Data Scarcity (4.2)	DT	[38]	✓	✓	Process	Independent	Real-time	Augmentation
		[39]	✓	✓	Process	Dependent	Non-real	Augmentation
	Transfer Learning	[40]		✓	Component	Dependent	Real-time	Augmentation
		[41]		✓	Component	Dependent	Real-time	Augmentation
		[42]		✓	Component	Independent	Non-real	Augmentation
		[43]		✓	Component	Dependent	Real-time	Augmentation
		[44]		✓	Process	Dependent	Non-real	Augmentation
	Indirect Estimation	[45]		✓	Component	Dependent	Non-real	Augmentation
		[46]		✓	Component	Dependent	Non-real	Augmentation
Insufficient Label (4.3)	Transfer Learning	[47]		✓	Component	Independent	Non-real	Augmentation
		[48]		✓	Component	Independent	Non-real	Augmentation
	AL	[50]		✓	Process	Independent	Non-real	Augmentation
		[51]		✓	Process	Independent	Non-real	Augmentation
		[52]		✓	Process	Independent	Non-real	Augmentation
	Semi-Supervised	[54]		✓	Component	Independent	Real-time	Augmentation
		[55]		✓	Component	Independent	Non-real	Augmentation

domain-specific knowledge. DTs provide the most powerful approach to tackling data imbalance and scarcity, owing to their ability to generate data in virtual environments. However, DTs typically require domain knowledge and additional resources to design and operate DT-based architectures.

Approaches to addressing data scarcity, including DTs, are predominantly domain-dependent. Transfer learning, when applied in data-scarce situations, often relies on information about domain similarity. Additionally, indirect estimation requires insight into the system model. If data from another domain are available, transfer learning becomes a promising approach. Conversely, if the system model is known, indirect estimation and DTs may yield the best results.

When it comes to addressing insufficient labels, an interesting observation is that all approaches are domain-independent. This independence stems from the fact that insufficient label research often leverages data-driven algorithms because of the availability of sufficient data despite the lack of labels. In this context, the promise of domain-specific approaches for insufficient labels is under-researched. Furthermore, most approaches focus on non-real-time augmentation. Implementing real-time automation in AI applications under insufficient label conditions is a very challenging task.

Figure 6 illustrates the challenges and solutions in the development phase of CPS-AI integration. To address the issue of drift, where the input data distribution or the relationship between model inputs and objectives changes, a proactive design can be implemented to anticipate potential drifts. In addition, AI models can be designed to detect and adapt to drift, or a strategy can be employed to periodically update the model in response to drift. To address data loss caused by network

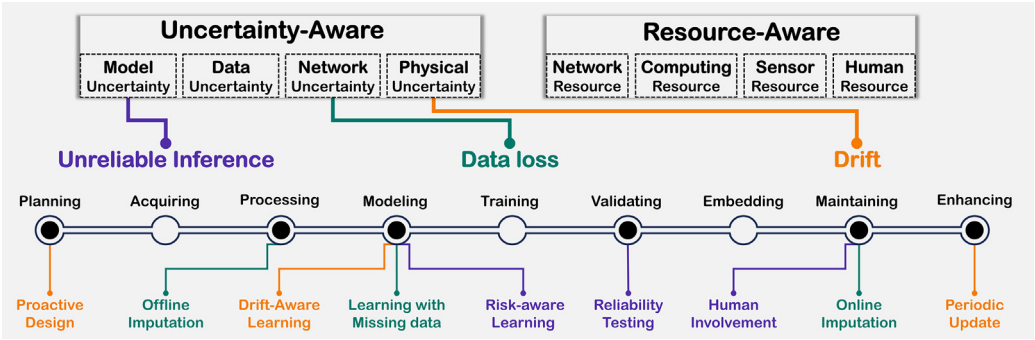

Fig. 6. Challenges and solutions in development phase.

uncertainties, imputation algorithms can be used to predict the missing values from both training and real-time data. Alternatively, models can be developed to train and infer from incomplete data. In the case of unreliable AI inferences, one solution is to develop AI models that recognize and mitigate the risks of these inferences. In addition, methods to validate the reliability of AI or strategies involving human intervention to prevent critical errors can be employed. The details of the challenges and solutions in the development phase are described in Sections 4.4–4.6.

4.4 Drift

One of the fundamental challenges in CPS-AI integration is that CPS inherently has dynamic uncertainty. In dynamically changing environments, data drift, where the distribution of input data changes over time, can occur. In addition, concept drift, where the relationship between model inputs and objectives changes, is also common. These drifts can occur abruptly, incrementally, gradually, or recurrently [56]. We refer to both concept drift and data drift as "drift." It is important to note that one-time anomalies are not considered drift. In CPS, such drift can result from physical system failures, parameter changes, network modifications, or external environmental changes, which can lead to unexpected outputs from AI models and cause severe errors or failures in the CPS.

4.4.1 Proactive Design. One way to address drift is to design for AI integration by predicting potential changes. This approach is particularly effective with a deep understanding of the target CPS. CPS often experiences concept drift due to frequent scenario changes in physical system operations. Bitsch and Schweitzer [57] propose a framework that automatically selects appropriate ML algorithms based on scenario information for **Automatic-Guided Vehicle (AGV)**. This framework defines the target parameters and execution deadlines that must be met according to the AGV's scenarios and identifies ML algorithms that can meet these requirements. Shen et al. [58] address the mathematical formalization of UAV swarm movements to develop a DT. By using data generated from the DT, they collect data for different scenarios to prevent significant performance degradation of control algorithms when scenarios change. Understanding the sources of noise can help anticipate and mitigate data drift. Bodo et al. [59] analyze potential sources of signal noise, bandwidth reduction, and computational cost in pump system monitoring. Based on this analysis, they investigate feature extraction methods that are robust to different noise scenarios.

4.4.2 Drift-Aware Learning. A well-known solution to address drift is to adopt adaptive learning, which detects and adapts to drift. Drift-aware learning was defined a decade ago by Joao Gama et al. [56]. A widely used approach in drift-aware learning is the ensemble method, where

multiple models perform prediction or classification on the same data. Lin et al. [60] address changing defect patterns in machine parts due to aging and maintenance/replacement using a Dynamic AdaBoost-based offline ensemble learning method. Kosana et al. [61] use nine Q-learning-based **Reinforcement Learning (RL)** models to effectively predict dynamically changing wind speeds in wind energy power grids. By continuously training the models based on performance and error metrics and selecting the optimal model in real-time, they adapt to the changing environment.

Another approach to drift-aware learning is the hybrid method, which combines different paradigms. Zhou et al. [62] address effective channel allocation for UAVs in scenarios where the number and location of UAVs change dynamically in real-time by combining LSTM and **Deep Q-Network (DQN)**. LSTM stores past behaviors and environmental states of UAVs, and its outputs are used as inputs to DQN, enabling rapid learning in dynamic environments. Jayaratne et al. [63] detect concept drift in unlabeled industrial CPS data by combining online k-means clustering with self-organizing map-based unsupervised learning, distinguishing between abrupt and recurring drifts. Bangui et al. [64] use a hybrid model that combines classification and clustering to respond to unexpected attacks in vehicular *ad hoc* networks. The classification model handles known attacks, while clustering addresses abnormal data. Che [65] proposes a hybrid approach to effectively predict AGV trajectories in environments with time-varying multi-sensor errors. This method uses error sequences and position data predicted by Autoregressive Integrated Moving Average as inputs to gradient boosting.

Recently, drift-aware model designs using the attention mechanism have been developed. The attention mechanism, which assigns weights to important parts of the input data, allows the model to focus on specific elements and is widely used in natural language processing. Kim et al. [66] propose a deep network based on the attention mechanism to solve the problems of non-periodic noise and sensitivity to input signal characteristics in sound-based machine fault diagnosis. The proposed model effectively controls the importance of phase components and input signal strength, making it generally applicable to various and complex types of machine sounds. Zhou et al. [67] present an AI-based traffic flow prediction model that effectively responds to spatio-temporal changes in data by using a CNN to extract the spatio-temporal dependencies of traffic data and employing the attention mechanism to dynamically adjust the importance of spatial features. Moon et al. [68] propose a graph-based **Deep RL (DRL)** method incorporating the attention mechanism to address the relationship and scale changes of tasks, machines, and vehicles in AGV-based flexible manufacturing systems.

4.4.3 Periodic Update. After an AI model is developed, it may be necessary to adapt it to drift. If the AI model itself lacks adaptability, a strategy of periodically updating the model during the enhancing step can be employed. Xu et al. [69] utilize GAN and DT to address new attacks or problem scenarios in CPS anomaly detection. They update the DT with real-time data and train the GAN based on the results predicted by the DT. The data obtained from the GAN are used to compute cross-entropy loss, which provides a direction for addressing drift. Bachinger et al. [70] propose an automated data versioning and model adaptation strategy for predictive models in smart manufacturing to adapt to changing conditions. This strategy tracks and manages the association of AI models with data versions, automatically adapting existing models to updated data versions or training entirely new models. Maschler et al. [71] propose Real2Sim and Sim2Real transfer learning to adapt AI-based control software to changing production requirements and system dynamics. The Real2Sim process reconfigures the simulation models according to new production requirements, while Sim2Real transfer learning transfers the ML algorithms trained by simulation models to the real system, reducing trial and error in the real system.

4.5 Data Loss

Due to its structure of interconnected cyber and physical systems through a network, CPS involves network uncertainties. In CPS using wireless networks, noise, collisions, unreliable links, and unexpected failures are major causes of data loss [72]. In addition, data loss can occur due to hardware failure in the physical system, software failures in the cyber system, and human error. Data loss during the training process can increase model bias and distortion, which ultimately degrades AI performance. Furthermore, data loss during the inference process can increase the data uncertainty and, depending on the conditions, lead to critical errors.

4.5.1 Offline Imputation. A common method for addressing data loss is to impute the missing values with predicted values, thereby creating a complete dataset. Data imputation involves estimating the most likely values that the actual sensors could have returned to deal with data loss [73]. Data imputation methods can be categorized into online imputation and offline imputation based on the timing of data processing [74]. Online imputation processes missing data as soon as it occurs, while offline imputation processes data using global information in a non-real-time manner. Data loss in training datasets is mainly handled during the processing step and is typically addressed in non-real-time operations.

Conventional data imputation methods widely used in statistics and data science can be employed. Cho et al. [32] address the loss of IoT data in manufacturing systems using methods such as KNN, linear interpolation, and multiple imputations by chained equation. The high-dimensional and timeseries nature of CPS data limits the effectiveness of conventional imputation methods. One way to address this is to use new AI models for data imputation. Zhou et al. [75] employ a hybrid method combining seasonal-trend decomposition with **Recurrent Neural Network (RNN)** to impute hydraulic and environmental monitoring data. Seasonal-trend decomposition is used to decompose long time series into trend and seasonality components to model long-term characteristics, while RNN models local patterns. Jeong et al. [76] propose a graph-based imputation method for high-dimensional data in chemical processes by defining similarities between independent variables, converting these similarities into a graph structure, and applying graph convolutional networks.

4.5.2 Learning with Missing Data. It is possible to use an approach that performs AI learning with missing data without data imputation [77]. Although learning with missing data has not received as much attention as data imputation, it is a crucial factor in preventing CPS failures. For this approach to be effective, it is necessary to extract important information from inputs with missing data and to make accurate inferences. To achieve this, [78] use a strategy of adding multiple time slices to a dynamic Bayesian network. They use an expectation-maximization optimization based on extensive historical data to make inferences using partially missing data and to detect machine errors.

LSTM, a type of RNN capable of retaining long-term information, is widely used. Premkumar et al. [79] propose a cyber-attack detection model that uses a combination of CNN and Bi-LSTM to extract important information from inputs, including missing data, and maintain model performance by using both past and future information. Muralidhar et al. [80] apply an attention mechanism to a bidirectional Seq2Seq architecture, a variant of LSTM, to predict the short- and long-term states of CPS. This approach is designed to deal with frequent failures in commercial hardware that result in periods of sparse or missing data in time series predictions for CPS. Some approaches perform imputation and training simultaneously, rather than training a model after data imputation. Yang et al. [81] propose a framework that jointly performs unsupervised learning and data imputation. They apply expectation-maximization-based optimization to both tasks. This method effectively

handles spatio-temporal dependencies between variables and demonstrates strong performance in anomaly detection in water treatment and chemical processes.

4.5.3 Online Imputation. Once the AI model is developed, data loss occurring during real-time inference cannot be addressed by offline imputation. Therefore, online data imputation (also known as real-time data imputation) is necessary to meet the time constraints of CPS. Llanes-Santiago [82] propose a framework for chemical anomaly detection using MLP with online data imputation to handle data loss. Advanced statistical techniques such as sequential regression, singular value decomposition, and local least squares estimation are employed for imputation. Sarda et al. [83] introduces a GAN-based data imputation method for real-time imputation of multivariate time series data in steel manufacturing processes. The GAN model is trained offline using collected timeseries data, and during the online phase, the model uses real-time data, including missing data, as input to generate imputed results.

4.6 Unreliable Inference

One of the greatest challenges to CPS-AI integration is the inherent uncertainty of AI. The intrinsic probabilistic properties and the limitations of algorithm design make AI's inference results unreliable. This increases risks for safety managers responsible for software-intensive CPS, as system malfunctions or errors can harm humans and the environment. To minimize these risks in CPS, several methods can be employed: (1) improve the quality of data-driven components, (2) prevent errors and clarify requirements, (3) perform integrated testing of combined systems, and (4) integrate humans into CPS [84]. We explore research aimed at minimizing the risks associated with the unreliable inferences of AI.

4.6.1 Risk-Aware Learning. To address unreliable inference, AI must be designed to consider the risks associated with AI inference. This approach is referred to as risk-aware learning [85, 86]. The primary focus of risk-aware learning is not to maximize the probability of the best inference but to avoid the worst inferences. In CPS, false positives can be critical in certain situations. Haghighi et al. [87] propose a conservative approach using classification and regression tree-based preventive rules. This method designs a classifier achieving a zero false-positive rate, thereby preventing the IDS of industrial systems from restricting legitimate traffic. Gu and Easwaran [88] minimize risk by inferring the uncertainty of the training data. They propose a strategy that partitions the feature space into smaller sections using a feature space partitioning tree. This strategy assesses the data distribution within each partition and, based on these evaluations, rejects inference in regions with insufficient training data.

By leveraging knowledge of physical systems, the risks associated with inference can be minimized. Zhou et al. [89] identify control actions that may pose risks in CPS, specify safe operations considering temporal constraints and signal characteristics, and use Signal Temporal Logic formulas to set a customized loss function that satisfies the safety context specifications. This approach develops a response time estimator that predicts the maximum time budget for the control software to take mitigating actions. Xu et al. [90] propose a RL training strategy that sets a **Maximum Absolute Error (MAE)** threshold that indicates control system stability and removes actions with a high bit error rate from the action space. This strategy ensures safe and fast convergence of RL through a policy of decreasing the exploration threshold as training progresses. Park et al. [91] propose adaptive RL learning in industrial wireless control systems to minimize the risk of packet loss. The strategy uses packets of lower importance for exploration to estimate packet loss rates and assigns packets of high importance for exploitation. Lyu et al. [92] propose an AI-enhancing framework that automatically adjusts deep learning controllers to meet safety requirements. The approach iteratively explores the control decision space to identify unsafe control actions, finds

optimal adjustments that satisfy the safety requirements, incorporates the adjusted control inputs and outputs into the training dataset, and retrains the deep learning controller.

4.6.2 Reliability Testing. Another way to minimize the risks of unreliable inferences is to test the reliability of the developed AI. Standard AI verification metrics such as accuracy, precision, and MSE are insufficient to verify reliability. Catak et al. [93] propose a validation technique to quantify uncertainty using a prediction-time dropout-based neural network and estimates the uncertainty of deep learning inferences using a support vector machine. Instead of setting predefined uncertainty thresholds, this approach effectively captures non-linear patterns in uncertainty through a prediction validation model. Song et al. [94] propose a stability evaluation benchmark for AI applied to CPS control, including goal achievement rate, MAE, safe state achievement time, and response time to events. Kim et al. [95] address the limitations of the F1 score in detecting anomalies from time series data, particularly its inability to indicate when an anomaly begins and ends. The proposed metrics, TaP (timeseries aware precision) and TaR (timeseries aware recall), use thresholds to distinguish values affected by anomalies during the return of the physical process to a normal state.

4.6.3 Human Involvement. One solution to the inherent uncertainty of AI is to involve humans in AI judgments and decisions. This involvement falls into two broad categories. One is the **Explainable AI (XAI)** paradigm, where the goal is to make AI inferences understandable to humans, enabling them to directly decide whether to accept or reject the AI's conclusions [96]. The other is the Safe Fail paradigm, where, in cases of high uncertainty in AI inferences, the option to reject the AI's inference is chosen, and traditional controllers or human operators apply backup solutions [88].

XAI approaches are explored to make AI inferences interpretable by users. To make decisions of DNN IDS interpretable by users, [97] use techniques such as RuleFit, which forms rule-based models, local interpretable model-agnostic explanations, which explain the reasons for predictions for specific inputs, and Shapley Additive Explanations, which distributes the impact of each feature on the prediction based on game theory. Similarly, [98] use Krill Herd Optimization, an optimization algorithm inspired by the swarming behavior of krill in nature, to improve the transparency of industrial CPS IDS. Each krill represents a feature of the dataset, and KHO evaluates the importance of the features, explaining which features the model considers in making predictions. Christou et al. [99] address the black-box problem of predicting the RUL of production systems by using quantitative association rule mining. QARMA generates rules to improve the transparency of AI by ranking the top 100 attribute values based on support and confidence.

The Safe Fail approach involves adding a process to determine whether to accept decisions based on uncertainty during the maintenance phase. Ramanagopal et al. [100] propose a method for identifying failures in deep learning-based object detection for autonomous vehicles, particularly when there is no ground truth data available. The method identifies failures of the object detector based on temporal cues and stereo cue mechanisms that analyze the movement and position of objects. Boursinos and Koutsoukos [101] present a system that assesses the reliability of outputs in deep learning-based recognition. This method derives confidence and credibility values p for each class, accepting the classification prediction if p exceeds a threshold, and involving human intervention if it does not.

Table 2 maps the studies presented in Sections 4.4–4.6 to the Three Dimensions of CPAI.

Most studies addressing challenges in the development phase primarily focus on component-level modifications. This is because these challenges significantly impact the inputs and outputs of AI models, leading to a prevalence of research that seeks to address them through the modification of AI components.

Table 2. Studies for Challenges of Development Phase

Challenge	Method	Ref.	Constraint		Approach		Purpose	
			Uncertainty-Awareness	Resource-Awareness	Scope of Change	Specificity of Domain	Timeliness	Responsibility
Drift (4.4)	Proactive Design	[57]	✓	✓	Process	Dependent	Real-time	Automation
		[58]	✓		Process	Dependent	Real-time	Automation
		[59]	✓	✓	Component	Dependent	Real-time	Augmentation
	Drift-Aware Learning	[60]	✓		Component	Independent	Non-real	Augmentation
		[61]	✓		Component	Independent	Real-time	Augmentation
		[62]	✓		Component	Independent	Real-time	Automation
		[63]	✓		Component	Independent	Real-time	Augmentation
		[64]	✓		Component	Independent	Real-time	Augmentation
		[65]	✓		Component	Dependent	Real-time	Augmentation
		[66]	✓		Component	Independent	Non-real	Augmentation
		[67]	✓		Component	Independent	Non-real	Augmentation
		[68]	✓		Component	Independent	Non-real	Automation
	Periodic Update	[69]	✓	✓	Process	Dependent	Real-time	Augmentation
		[70]	✓		Process	Independent	Real-time	Augmentation
		[71]	✓		Process	Dependent	Real-time	Automation
Data Loss (4.5)	Offline Imputation	[32]	✓		Component	Independent	Non-real	Augmentation
		[75]	✓		Component	Independent	Non-real	Augmentation
		[76]	✓		Component	Independent	Non-real	Augmentation
	Learning with Missing Data	[78]	✓		Component	Dependent	Real-time	Augmentation
		[79]	✓	✓	Component	Independent	Non-real	Augmentation
		[80]	✓		Component	Dependent	Non-real	Augmentation
		[81]	✓		Component	Dependent	Non-real	Augmentation
	Online Imputation	[82]	✓	✓	Component	Independent	Real-time	Augmentation
		[83]	✓	✓	Component	Independent	Real-time	Augmentation
Unreliable Inference (4.6)	Risk-Aware Learning	[87]	✓		Component	Independent	Non-real	Augmentation
		[88]	✓		Component	Independent	Non-real	Augmentation
		[89]	✓	✓	Component	Independent	Real-time	Automation
		[90]	✓	✓	Component	Dependent	Real-time	Automation
		[91]	✓		Component	Dependent	Real-time	Automation
		[92]	✓	✓	Component	Independent	Real-time	Automation
	Reliability Testing	[93]	✓		Component	Independent	Non-real	Augmentation
		[94]	✓		Process	Independent	Non-real	Augmentation
		[95]	✓		Component	Independent	Real-time	Automation
	Human Involvement	[97]	✓		Component	Independent	Real-time	Automation
		[98]	✓		Component	Independent	Real-time	Automation
		[99]	✓		Component	Independent	Real-time	Augmentation
		[100]	✓		Component	Dependent	Real-time	Augmentation
		[101]	✓		Component	Dependent	Non-real	Augmentation

Solutions to drift have been extensively studied in the context of real-time applications, where drift is particularly critical. Designing drift-aware AI models is a common solution to the drift problem. This approach emphasizes the creation of AI models that recognize and adapt to drift based on data, making it domain-independent. In addition, it requires modifications only to AI components, which minimizes implementation constraints. In contrast, proactive design anticipates drift based on information about the system, making it domain-dependent. However, it does not require the design of new AI models and is relatively more robust to drift compared to drift-aware strategies. Periodic updates have the advantage of being applicable even after the AI model is embedded in the system. However, unlike other methods, this approach requires process-level changes, such as the use of DTs, simulators, or data management systems, which involve additional resources.

A common solution to data loss is to incorporate an imputation function into the data processing component. Offline imputation is performed using collected data and is thus suited for non-real-time applications. In contrast, online imputation is required for real-time applications, as it addresses

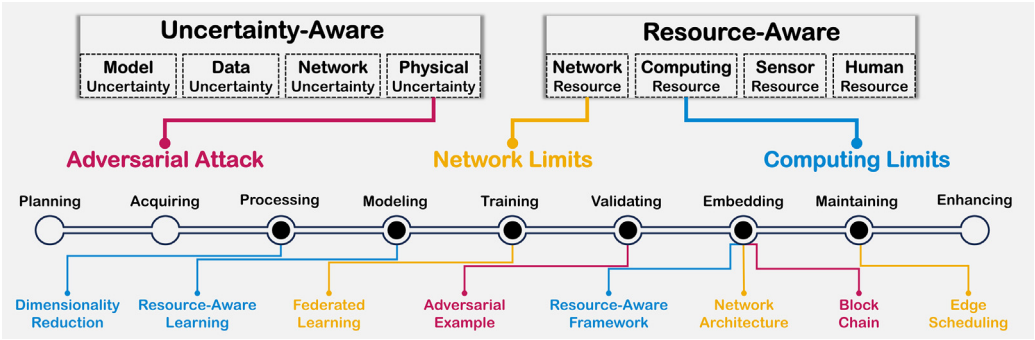

Fig. 7. Challenges and solutions in deployment phase.

data loss using real-time series data. Additionally, the learning with missing data approach involves redesigning AI models and often requires domain knowledge, as it focuses on extracting critical features from timeseries data. We found no studies addressing data loss in applications aimed at automation. Addressing the data loss challenge in the context of automation would be an intriguing area for future research.

Unreliable inference has been addressed more extensively in real-time automation applications compared to other challenges. This is because uncertain inference poses one of the most critical threats to the responsibility of AI systems. Risk-aware learning involves designing AI models to avoid the worst-case inference scenarios. It can utilize domain-independent rejection strategies or domain-specific inference formulation strategies. However, such approaches do not guarantee optimal performance. Human involvement approaches rely on human intervention when inference fails. While these approaches preserve AI performance better than risk-aware learning, their application is limited due to their dependency on human input. Reliability testing, on the other hand, focuses on conducting additional validation for trained AI models. This method does not require designing new models but is not a fundamental solution to the problem. Furthermore, the testing process may involve additional procedures, such as evaluating invalidated models within CPS.

Figure 7 illustrates the challenges and solutions in the deployment phase of CPS-AI integration. To address computing resource limits, reducing the dimensionality of the high-dimensional data of the CPS can be a solution. In addition, resource-aware learning, which adapts AI model structures or training methods to CPS resources, and resource-aware frameworks, which transform CPS frameworks to accommodate AI, are being explored. Similarly, to tackle network resource limits, designing the network architecture of the environment where AI is embedded is studied. In addition, federated learning strategies can reduce data transmission, and the scheduling of data transmission at the edge can be optimized. Finally, similar to other components of CPS, AI may be subject to adversarial attacks. To counter this, examples of such attacks can be included in the training step, or blockchain mechanisms can be integrated into the AI operational processes. The details of the challenges and solutions in the deployment phase are described in Sections 4.7–4.9.

4.7 Computing Limits

In CPS, computing resource constraints must be considered due to the limited computing power available. In particular, AI, with its complex model structures, large datasets, and high computational demands, consumes significantly more computing resources than traditional methods. In CPS-AI integration, these limited computing resources can make AI training infeasible or result in excessive

time and energy consumption, negatively impacting other CPS processes. Computing limits can be addressed through two main approaches: process-level changes that adapt the CPS architecture to the resources of the AI and component-level changes that adapt the data and AI models to the resources of the CPS.

4.7.1 Resource-Aware Framework. Designing a resource-aware framework that adapts the CPS architecture to the resource consumption of the AI is one of the solutions to computing limits. First, the framework can be designed by analyzing the resources consumed by the AI. Trihinas et al. [102] propose a framework to monitor the energy usage of UAVs with ML applications to prevent errors caused by energy overhead. This framework measures the energy associated with flight, computation, and communication based on on-board drone data and then generates an energy model formulated through equations. Dhuheir et al. [103] define constraints on computational load and maximum memory usage for UAV swarms to eliminate decision delays caused by resource constraints in on-board image classification ML. This approach prevents ML applications from causing resource exhaustion. Agyeman and Rinner [104] address local resource constraints where the resource requirements of ML are significantly exceed the resources of smart cameras. They propose resource-appropriate data input/output, data volume, data compression methods, and distributed processing methods.

To overcome the limits of on-board computing resources, approaches using DTs are also being explored. Lei et al. [105] address the difficulty of directly utilizing ML in UAV swarms due to computational and storage constraints by using intelligent centers. In this approach, DTs are constructed at the intelligent centers for data collection and model training, and then the trained models are embedded in the UAVs to supplement their limited computational and storage capabilities. Similarly, [106] use a task allocation algorithm with DQN in UAV swarms by conducting pre-training through DTs on an airship equipped with a high-performance computer. Subsequently, real-time interactions between the UAVs and the DT ensure the validity of the pre-training results by adjusting the operational and state spaces in real-time.

4.7.2 Dimensionality Reduction. The high dimensionality of data used in AI complicates models and increases computational requirements. In the processing step, dimensionality reduction of data generated by CPS is one way to address the constraints on computing limits. Common methods such as **Principal Component Analysis (PCA)** and Kernel PCA are often ineffective for complex CPS data. Therefore, new feature selection or extraction algorithms suitable for CPS data need to be used. Bodo et al. [59] propose a strategy that ranks the features based on industrial constraints such as environmental conditions, sensor types, and the computation time of feature extraction algorithms. Tertytchny and Michael [107] also suggest data reduction through feature evaluation and ranking based on information gain. Tutsoy et al. [108] apply RL to humanoid robots, reducing the dimensionality of the state space by using symbolic inverse kinematics. They derive explicit equations for each joint to reduce the dimensionality of the state space.

Recently, there have been increasing attempts to use AI for dimensionality reduction. Ghahramani et al. [109] propose a feature selection method based on genetic algorithms and **Artificial Neural Networks (ANN)** to effectively recognize non-linear relationships in about 600 high-dimensional semiconductor process data. Zhong et al. [110] employ a parameter extraction method combining ANN and Pearson correlation coefficient to effectively capture spatio-temporal correlations in high-dimensional UAV flight data. Mansour [111] employs a meta-heuristic algorithm, adaptive harmony search, for feature selection in high-dimensional data of CPS IDS.

4.7.3 Resource-Aware Learning. Computing limits can be addressed by modifying the structure or training methods of AI models to suit the computing resources of the CPS. Kiac et al. [112] reduce

computational complexity by reducing the number of filters in convolutional layers and removing unnecessary blocks when using AI to recognize real-time traffic conditions at intersections and railway crossings. Li et al. [113] address the computational burden in UAV trajectory planning using DRL by modifying the initial DRL training process. They use structured weight pruning to retain only important connections while removing unnecessary or less important neurons and connections, thereby reducing memory usage and increasing processing speed.

One of the representative approaches is to maximize the use of computing resources on edge devices. Ren et al. [114] utilize an adaptive knowledge distillation algorithm to deploy a neural network model for RUL prediction on edge devices. The knowledge distillation algorithm transfers the knowledge from a high-performance teacher model to a lightweight student model, enabling the smaller model to mimic the performance of the larger model. Jin et al. [115] propose an edge-based cooperative learning system to maximize learning performance while using resources efficiently during training at the edge. The parameter server receives local gradients to update and store model parameters, and edge devices train the model with their data. During this process, the system evaluates the status of each device in real-time to determine the optimal task order for network and computational resources.

Some studies explore performing federated learning on edge devices with limited computing resources. For more details on federated learning, see Section 4.8.1. Mills et al. [116] address the issue of computational cost in IoT device-based federated learning by using distributed Adam optimization and model compression methods to reduce the time required for model convergence. Jiang et al. [117] apply federated learning to CPS with limited computing resources by using a parameter selection method based on update direction consistency.

4.8 Network Limits

One of the key characteristics of CPS is that all data from cyber and physical systems are transmitted over the network. Since network resources are not infinite, the amount of data that can be transmitted in real-time is limited. This limitation is even more pronounced when using wireless networks. Network limits can make AI training and inference impossible, and AI-related traffic can interfere with the transmission of data for other processes. To address network limits, AI models or network architectures can be modified. In addition, network scheduling can be designed with AI in mind.

4.8.1 Federated Learning. Distributed learning improves computational efficiency but reduces network efficiency due to the process of distributing data from a central server to each device. Federated learning is a learning method that can address the network constraints in distributed learning. In federated learning, models are trained on multiple local devices and then only the learned parameters are uploaded to a central server to improve the overall system model. Since the actual data are not transmitted over the network, this reduces network resource consumption and is more secure from an information protection perspective. Recently, there have been increasing attempts to use federated learning in CPS. [118] proposes federated feature selection in the distributed computing of autonomous vehicles. Local feature selection is performed using cross-entropy, and the central server iterates rounds using a Bayesian approach until the global probability vector stabilizes.

Federated learning is most widely used in CPS cybersecurity. Jayasinghe et al. [119] propose a federated learning-based hierarchical anomaly detection mechanism. This mechanism uses two anomaly detectors: the first uses a relatively small database with an ML model and a local aggregation server, while the second one uses a more complex ML model and a global aggregation server for federated learning. For intrusion detection, [120] use a **Gated Recurrent Unit (GRU)**

model, [121] combine CNN and GRU models, and [122] combine CNN and MLP models. These approaches use federated learning to address security threats and network overhead associated with transmitted data.

4.8.2 Network Architecture. Transforming the network architecture of CPS to accommodate AI can address network limitations. Yun et al. [123] propose an importance-based intelligence-defined networking architecture that assigns different weights to dataflows based on their importance level. The importance of dataflows is classified through clustering, and the weights are used to quantify the performance score of DNN models to evaluate resource allocation performance.

Network architecture is particularly noteworthy from the perspective of distributed learning, where network efficiency is crucial. Lim et al. [124] propose a mechanism using UAVs to address network delays and failures caused by network range limitations in federated learning of IoT devices. After local devices perform model training, UAVs move to points with favorable channel conditions for an efficient wireless network, collect updated model parameters and circulate them through the network to increase network coverage. Shen [125] propose an adaptive framework to resolve the differences in computation and transmission speed between local devices when using distributed deep learning in heterogeneous environments. The framework evaluates the learning time of devices, uses a hybrid of asynchronous and semi-synchronous communication, and employs data compression strategies during learning. Zhao et al. [126] address the single-point failure problem in IIoT federated learning by proposing a model aggregation structure with multiple servers. The strategy minimizes the loss of the central model by considering potential failures during data transmission and distributing the data to multiple servers.

4.8.3 Edge Scheduling. To deploy AI with limited computing resources, edge computing architectures that leverage the computational resources of edge devices are widely used. These edge computing architectures create additional processes for assigning tasks to distributed edge devices and transmitting data. Edge scheduling is required to address the network latency resulting from this process. Sun et al. [127] propose a scheduling method to solve the network delay issue in real-time IIoT by combining edge computing and cloud computing. This method distributes network traffic between edge servers and remote cloud servers. The scheduling algorithm determines task offloading and traffic routing based on the performance of AI and the estimated latency for each edge server and cloud. Li et al. [128] propose a scheduling method using SDN to guarantee real-time performance by minimizing delays in edge computing-based services for smart manufacturing. This method evaluates the state and waiting time of edge devices to appropriately select the server that will execute the AI model.

Ihekoronye et al. [129] propose a scheduling method to minimize latency and maximize throughput in mobile edge computing-based UAV intrusion detection. The strategy uses MEC-UAVs, which act as gateways between the UAV swarm and the GCS, delivering control signals and offloading high-cost computational tasks. Zhu et al. [130] use RL to minimize the energy consumption of edge devices. DRL scheduling, which considers the energy consumption and processing latency (including network and transmission delays) of AI tasks performed at the edge, distributes tasks to edge devices. Khan et al. [131] formulate the federated learning process of IoT devices as an integer linear optimization problem to achieve resource optimization. The formulated problem, considering hardware limitations, packet error rates, and interference, is decomposed into sub-problems of association and resource allocation, which are then solved using convex optimization techniques.

4.9 Adversarial Attack

In general, CPS have employed oracle-based mechanisms to detect abnormal attacks [132] and have further enhanced these capabilities through the integration of AI into the oracle design [133].

However, with the emergence of adversarial attacks aimed at deceiving AI in CPS, new strategies are required to effectively counteract such threats. Adversarial attacks on AI involve generating intentionally perturbed inputs to cause the AI to make incorrect decisions [134]. These adversarial attacks can be considered as a form of data drift. Current research on these attacks remains focused on the cyber domain, leaving the potential risks posed by adversarial examples to CPS largely unexplored [135].

4.9.1 Adversarial Example. One approach to counter adversarial attacks is to anticipate such attacks during the training and validation process. Pattanaik et al. [136] propose a strategy to counteract adversarial attacks during the training step. To address adversarial attacks in the use of RL in control systems, adversarial training is incorporated into the DRL model training process to improve the robustness to parameter uncertainties in the DRL algorithm. Another solution is to address adversarial attacks during the validation step. Pet et al. [137] introduce the DeepXplore testing technique, which measures the neuron activation coverage of deep learning models in autonomous driving by adding adversarial test inputs that induce differential behaviors to traditional random testing. Zhou et al. [138] test the robustness of deep learning-based anomaly detection models in CPS by incorporating Gaussian-based noise and adversarial examples generated by neural networks during the validation process.

4.9.2 Block Chain. One solution to counter adversarial attacks is to use blockchain technology. Blockchain is a distributed digital ledger technology that stores data in immutable blocks, each linked to the previous one. This ensures data integrity and prevents hacking and data tampering. CPS data and commands stored on the blockchain prevent intentional data manipulation and facilitate the detection and tracing of attack attempts. Wang et al. [139] add a blockchain-based process and noise injection feature during the model update process to prevent vulnerabilities in the central server and malicious local data provisioning when using federated learning-based crowd detection ML in UAVs. Similarly, [140] address security issues in federated learning-based anomaly detection in IoT by applying controllable noise and blockchain to local parameters sent to the central server. Ren et al. [141] propose an NFT-based intelligence networking system in a sensor data collection system through automated vehicles. The roadside unit stores hashes via blockchain that individual vehicles can download and use. Singh et al. [142] propose a decentralized IoT architecture using blockchain. This four-layer structure, consisting of edge, base station, fog, and cloud, connects each layer to the blockchain to run intelligent applications.

Table 3 maps the studies introduced in Sections 4.7–4.9 to the Three Dimensions of CPAI.

Most studies on deployment-phase challenges involve considering resource constraints, as these challenges significantly affect the operational environment in which AI components run.

Computing limits have been extensively studied due to their profound impact on running AI under real-time conditions. A common solution to computing limits is to adopt resource-aware frameworks and apply dimensionality reduction methods. Resource-aware frameworks require process-level changes but do not necessitate modifying the AI model itself. They offer a practical solution by coordinating resource usage between the CPS and the AI, ensuring more efficient utilization of available resources. In contrast, dimensionality reduction only requires modifications at the feature level of the AI model and is generally domain-independent, making it compatible with other approaches. However, simplifying input data can lead to some loss in performance. Resource-aware learning represents adjusting the AI model itself to accommodate resource constraints. This can be achieved by reducing computational overhead within a single model or by optimizing resource usage across multiple models in distributed or federated learning scenarios. While these approaches are domain-independent, it is important to choose the appropriate learning strategy based on the operational environment.

Table 3. Studies for Challenges of Deployment Phase

Challenge	Method	Ref.	Constraint		Approach		Purpose	
			Uncertainty-Awareness	Resource-Awareness	Scope of Change	Specificity of Domain	Timeliness	Responsibility
Computing Limits (4.7)	Resource-Aware Framework	[102]		✓	Process	Dependent	Real-time	Augmentation
		[103]		✓	Process	Dependent	Real-time	Augmentation
		[104]		✓	Process	Independent	Real-time	Augmentation
		[105]		✓	Process	Dependent	Real-time	Automation
		[106]		✓	Process	Dependent	Real-time	Automation
	Dimensionality Reduction	[59]	✓	✓	Component	Dependent	Real-time	Augmentation
		[107]		✓	Component	Independent	Non-real	Augmentation
		[108]		✓	Component	Dependent	Real-time	Automation
		[109]		✓	Component	Independent	Non-real	Augmentation
		[110]		✓	Component	Independent	Non-real	Augmentation
		[111]		✓	Component	Independent	Non-real	Augmentation
	Resource-Aware Learning	[112]		✓	Component	Independent	Real-time	Augmentation
		[113]		✓	Component	Independent	Real-time	Automation
		[114]		✓	Process	Independent	Non-real	Augmentation
		[115]		✓	Process	Independent	Non-real	Augmentation
		[116]	✓	✓	Component	Independent	Non-real	Augmentation
		[117]	✓	✓	Component	Independent	Non-real	Augmentation
Network Limits (4.8)	Federated Learning	[118]	✓	✓	Component	Independent	Non-real	Augmentation
		[119]	✓	✓	Process	Independent	Non-real	Augmentation
		[120]	✓	✓	Process	Independent	Real-time	Augmentation
		[121]	✓	✓	Process	Dependent	Non-real	Augmentation
		[122]	✓	✓	Process	Independent	Non-real	Augmentation
	Network Architecture	[123]		✓	Process	Independent	Real-time	Augmentation
		[124]		✓	Process	Dependent	Real-time	Augmentation
		[125]		✓	Process	Dependent	Real-time	Automation
		[126]		✓	Component	Independent	Real-time	Automation
	Edge Scheduling	[127]		✓	Process	Independent	Real-time	Augmentation
		[128]		✓	Component	Independent	Real-time	Automation
		[129]		✓	Process	Dependent	Real-time	Augmentation
		[130]		✓	Process	Dependent	Real-time	Augmentation
		[131]		✓	Process	Dependent	Real-time	Augmentation
Adversarial Attack (4.9)	Adversarial Example	[136]	✓		Component	Independent	Real-time	Automation
		[137]	✓	✓	Component	Independent	Real-time	Automation
		[138]	✓	✓	Component	Dependent	Non-real	Augmentation
	Block Chain	[139]	✓	✓	Component	Dependent	Non-real	Augmentation
		[140]	✓	✓	Component	Independent	Non-real	Augmentation
		[141]	✓	✓	Component	Dependent	Real-time	Automation
		[142]	✓	✓	Process	Independent	Real-time	Augmentation

Within the context of network limits, federated learning has recently gained attention as it can address both network constraints and adversarial uncertainties. However, implementing federated learning may require substantial process-level changes in non-distributed operational environments. Additionally, ensuring model synchronization in real-time scenarios can be challenging. Another strategy to address network constraints is to adjust the CPS network architecture. Such adjustments are generally undertaken with the assumption of distributed or federated learning, which may demand additional AI model or computational resources during the network optimization process. Edge scheduling mitigates the additional processes introduced by federated learning and network architecture adjustments. Although edge scheduling has been extensively studied in real-time systems, it demands substantial domain knowledge to simultaneously consider the unique characteristics of AI workloads and the CPS network.

The challenge of adversarial attacks remains an area that requires significant further research. Currently, a common solution is to include adversarial examples in the training process of AI models. While this approach is effective, it increases computational overhead and does not provide

absolute protection. Another solution is the use of blockchain technology, which can enhance system integrity by integrating blockchain-based processes into systems. This method is relatively straightforward to implement but introduces additional computational and storage requirements. We believe that risk-aware learning and drift-aware learning approaches hold promise for addressing adversarial attack challenges. This is because adversarial attacks can be considered a form of drift that often results in unreliable inference.

5 Conclusion

CPS-AI integration presents significant opportunities and formidable challenges. AI operates on the premise that "good things happen probabilistically," while CPS adheres to the principle that "bad things must not happen." This fundamental difference requires addressing and resolving the uncertainties inherent in CPS-AI integration. Representative challenges include data imbalance, data loss, drift, unreliable inference, and adversarial attacks. Furthermore, the clash between the AI academic perspective that "resources are always accessible and AI-centric" and the CPS principle emphasizing that "resources are inherently limited and shared among all components" amplifies the importance of resource-efficient design in CPS-AI integration. Typical resource-related challenges include data and label scarcity, network limitations, and computational constraints.

We introduce CPAI, a specialized sub-field within AI research which we propose as the first of its kind. CPAI encompasses various technologies and methodologies aimed at addressing these constraints. To clarify the scope and definition of CPAI, we provide a novel 3D classification scheme consisting of Constraint (C), Purpose (P), and Approach (A). Additionally, we reevaluate 104 studies from a CPAI perspective, highlighting key challenges and insights from a CPAI perspective. Through this article, we aim to consolidate fragmented research efforts in the emerging CPAI domain, fostering collaboration and progress among CPS and AI researchers. This integrated approach not only facilitates the practical application of AI in CPS but also guides researchers in developing resource-efficient and reliable AI technologies that meet the stringent requirements of CPS. By introducing the CPAI domain, we hope to ultimately enable the reliable and resource-efficient design, development, and deployment of AI technologies in CPS environments.

References

[1] Kyung-Joon Park, Rong Zheng, and Xue Liu. 2012. Cyber-physical systems: Milestones and research challenges. *Computer Communications* 36, 1 (2012), 1–7.

[2] Sangjun Kim and Kyung-Joon Park. 2021. A survey on machine-learning based security design for cyber-physical systems. *Applied Sciences* 11, 12 (2021), 5458.

[3] Jiyeong Chae, Sanghoon Lee, Junhyung Jang, Seohyung Hong, and Kyung-Joon Park. 2023. A survey and perspective on industrial cyber-physical systems (ICPS): From ICPS to AI-augmented ICPS. *IEEE Transactions on Industrial Cyber-Physical Systems* 1 (2023), 257–272.

[4] Eric M. S. P. Veith, Lars Fischer, Martin Tröschel, and Astrid Nieße. 2020. Analyzing cyber-physical systems from the perspective of artificial intelligence. In *Proceedings of the 2019 International Conference on Artificial Intelligence, Robotics and Control (AIRC '19)*. ACM, New York, NY, 85–95.

[5] Jan Kocoń, Igor Cichecki, Oliwier Kaszyca, Mateusz Kochanek, Dominika Szydło, Joanna Baran, Julita Bielaniewicz, Marcin Gruza, Arkadiusz Janz, Kamil Kanclerz, et al. 2023. ChatGPT: Jack of all trades, master of none. *Information Fusion* 99 (2023), 101861.

[6] Sanghoon Lee, Jinyoung Kim, Gwangjin Wi, Yuchang Won, Yongsoon Eun, and Kyung-Joon Park. 2024. Deep reinforcement learning-driven scheduling in multijob serial lines: A case study in automotive parts assembly. *IEEE Transactions on Industrial Informatics* 20, 2 (2024), 2932–2943.

[7] Victor Bolbot, Gerasimos Theotokatos, Luminita Manuela Bujorianu, Evangelos Boulougouris, and Dracos Vassalos. 2019. Vulnerabilities and safety assurance methods in cyber-physical systems: A comprehensive review. *Reliability Engineering & System Safety* 182 (2019), 179–193.

[8] Zhenge Jia, Jianxu Chen, Xiaowei Xu, John Kheir, Jingtong Hu, Han Xiao, Sui Peng, Xiaobo Sharon Hu, Danny Chen, and Yiyu Shi. 2023. The importance of resource awareness in artificial intelligence for healthcare. *Nature Machine Intelligence* 5, 7 (2023), 687–698.

[9] Dohwan Kim, Yuchang Won, Seunghyeon Kim, Yongsoon Eun, Kyung-Joon Park, and Karl H. Johansson. 2019. Sampling rate optimization for IEEE 802.11 wireless control systems. In *Proceedings of the 10th ACM/IEEE International Conference on Cyber-Physical Systems (ICCPS '19)*. ACM, New York, NY, 87–96.

[10] Sihoon Moon, Sanghoon Lee, Wonhong Jeon, and Kyung-Joon Park. 2023. Learning-enabled network-control co-design for energy-efficient industrial Internet of things. *IEEE Transactions on Network and Service Management* 1, 1 (2023), 1–13.

[11] Zhao Li, Chengcheng Huang, Xiaoxiao Dong, and Chongguang Ren. 2020. Resource-efficient cyber-physical systems design: A survey. *Microprocessors and Microsystems* 77 (2020), 103183.

[12] Emma Strubell, Ananya Ganesh, and Andrew McCallum. 2020. Energy and policy considerations for modern deep learning research. *Proceedings of the AAAI Conference on Artificial Intelligence* 34, 9 (Apr. 2020), 13693–13696.

[13] Yanzhi Wang, Caiwen Ding, Zhe Li, Geng Yuan, Siyu Liao, Xiaolong Ma, Bo Yuan, Xuehai Qian, Jian Tang, Qinru Qiu, et al. 2018. Towards ultra-high performance and energy efficiency of deep learning systems: An algorithm-hardware co-optimization framework. *Proceedings of the AAAI Conference on Artificial Intelligence* 32, 1 (Apr. 2018), 1–9.

[14] Luis M. C. Oliveira, Rafael Dias, Carine M. Rebello, Márcio A. F. Martins, Alírio E. Rodrigues, Ana M. Ribeiro, and Idelfonso B. R. Nogueira. 2021. Artificial intelligence and cyber-physical systems: A review and perspectives for the future in the chemical industry. *AI* 2, 3 (2021), 27.

[15] Euiyoung Chung. 2022. Domain knowledge-based human capital strategy in manufacturing AI. *IEEE Engineering Management Review* 51, 1 (2022), 108–122.

[16] Mark Ryan. 2023. The social and ethical impacts of artificial intelligence in agriculture: Mapping the agricultural AI literature. *AI & Society* 38, 6 (2023), 2473–2485.

[17] Alshaibi Ahmed Jamal, Al-Ani Mustafa Majid, Anton Konev, Tatiana Kosachenko, and Alexander Shelupanov. 2023. A review on security analysis of cyber physical systems using machine learning. *Materials Today: Proceedings* 80 (2023), 2302–2306.

[18] Jun Zhang, Lei Pan, Qing-Long Han, Chao Chen, Sheng Wen, and Yang Xiang. 2021. Deep learning based attack detection for cyber-physical system cybersecurity: A survey. *IEEE/CAA Journal of Automatica Sinica* 9, 3 (2021), 377–391.

[19] Rajesh H Kulkarni and Palacholla Padmanabham. 2017. Integration of artificial intelligence activities in software development processes and measuring effectiveness of integration. *IET Software* 11, 1 (2017), 18–26.

[20] Maik Frye, Johannes Mohren, and Robert H. Schmitt. 2021. Benchmarking of data preprocessing methods for machine learning-applications in production. *Procedia CIRP* 104 (2021), 50–55.

[21] Dragutin Petkovic. 2023. It is not "Accuracy vs. Explainability"—we need both for trustworthy AI systems. *IEEE Transactions on Technology and Society* 4, 1 (2023), 46–53.

[22] Elisa Negri, Luca Fumagalli, and Marco Macchi. 2017. A review of the roles of digital twin in CPS-based production systems. *Procedia Manufacturing* 11 (2017), 939–948.

[23] Andrea Castellani, Sebastian Schmitt, and Stefano Squartini. 2020. Real-world anomaly detection by using digital twin systems and weakly supervised learning. *IEEE Transactions on Industrial Informatics* 17, 7 (2020), 4733–4742.

[24] Artur Liebert, Christian Wittke, Jonas Ehrhardt, Richard Jaufmann, Niklas Widulle, Sebastian Eilermann, Maria Krantz, and Oliver Niggemann. 2023. Using FliPSi to generate data for machine learning algorithms. In *Proceedings of the 2023 IEEE 28th International Conference on Emerging Technologies and Factory Automation (ETFA)*. IEEE, 1–8.

[25] Tijana Markovic, Miguel Leon, Bjorn Leander, and Sasikumar Punnekkat. 2023. A modular ice cream factory dataset on anomalies in sensors to support machine learning research in manufacturing systems. *IEEE Access* 11 (2023), 29744–29758.

[26] Qinghua Xu, Shaukat Ali, Tao Yue, Nedim Zaimovic, and Inderjeet Singh. 2023. KDDT: knowledge distillation-empowered digital twin for anomaly detection. In *Proceedings of the 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE)*, 1867–1878.

[27] Julio Galvan, Ashok Raja, Yanyan Li, and Jiawei Yuan. 2021. Sensor data-driven UAV anomaly detection using deep learning approach. In *Proceedings of the 2021 IEEE Military Communications Conference (MILCOM)*. IEEE, 589–594.

[28] Markus Bayer, Marc-André Kaufhold, and Christian Reuter. 2022. A survey on data augmentation for text classification. *ACM Computing Surveys* 55, 7 (2022), 1–39.

[29] Alhassan Mumuni and Fuseini Mumuni. 2022. Data augmentation: A comprehensive survey of modern approaches. *Array* 16 (2022), 100258.

[30] Fátima A. Saiz, Garazi Alfaro, Iñigo Barandiaran, and Manuel Graña. 2021. Generative adversarial networks to improve the robustness of visual defect segmentation by semantic networks in manufacturing components. *Applied Sciences* 11, 14 (2021), 6368.

[31] Sanghoon Lee, Jiyeong Chae, Sihoon Moon, Sang-Chul Lee, and Kyung-Joon Park. 2024. False alarm prevention through domain knowledge-driven machine learning: Leakage detection in water distribution networks. *IEEE Sensors Journal* 24, 19 (2024), 31538–31550.

[32] Eunnuri Cho, Tai-Woo Chang, and Gyusun Hwang. 2022. Data preprocessing combination to improve the performance of quality classification in the manufacturing process. *Electronics* 11, 3 (2022), 477.

[33] Hadi Hojjati and Narges Armanfard. 2022. Self-supervised acoustic anomaly detection via contrastive learning. In *Proceedings of the 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 3253–3257.

[34] Hoang V. Vo, Hanh P. Du, and Hoa N. Nguyen. 2023. AI-powered intrusion detection in large-scale traffic networks based on flow sensing strategy and parallel deep analysis. *Journal of Network and Computer Applications* 220 (2023), 103735.

[35] Jeongsu Lee, Young Chul Lee, and Jeong Tae Kim. 2020. Fault detection based on one-class deep learning for manufacturing applications limited to an imbalanced database. *Journal of Manufacturing Systems* 57 (2020), 357–366.

[36] Kaixiang Yang, Yifan Shi, Zhiwen Yu, Qinmin Yang, Arun Kumar Sangaiah, and Huanqiang Zeng. 2022. Stacked one-class broad learning system for intrusion detection in industry 4.0. *IEEE Transactions on Industrial Informatics* 19, 1 (2022), 251–260.

[37] Xueming Ye, Xing Wu, and Yike Guo. 2018. Real-time quality prediction of casting billet based on random forest algorithm. In *Proceedings of the 2018 IEEE International Conference on Progress in Informatics and Computing (PIC)*. IEEE, 140–143.

[38] Kosmas Alexopoulos, Nikolaos Nikolakis, and George Chryssolouris. 2020. Digital twin-driven supervised machine learning for the development of artificial intelligence applications in manufacturing. *International Journal of Computer Integrated Manufacturing* 33, 5 (2020), 429–439.

[39] Weichao Luo, Tianliang Hu, Yingxin Ye, Chengrui Zhang, and Yongli Wei. 2020. A hybrid predictive maintenance approach for CNC machine tool driven by digital twin. *Robotics and Computer-Integrated Manufacturing* 65 (2020), 101974.

[40] Shaohua Huang, Yu Guo, Daoyuan Liu, Shanshan Zha, and Weiguang Fang. 2019. A two-stage transfer learning-based deep learning approach for production progress prediction in IoT-enabled manufacturing. *IEEE Internet of Things Journal* 6, 6 (2019), 10627–10638.

[41] Changchun Liu, Haihua Zhu, Dunbing Tang, Qingwei Nie, Shipei Li, Yi Zhang, and Xuan Liu. 2023. A transfer learning CNN-LSTM network-based production progress prediction approach in IIoT-enabled manufacturing. *International Journal of Production Research* 61, 12 (2023), 4045–4068.

[42] Peng Wang and Robert X. Gao. 2020. Transfer learning for enhanced machine fault diagnosis in manufacturing. *CIRP Annals* 69, 1 (2020), 413–416.

[43] Huan Wang, Haifeng Zhang, Lei Zhu, Yan Wang, and Junyi Deng. 2023. ResADM: A transfer-learning-based attack detection method for cyber–physical systems. *Applied Sciences* 13, 24 (2023).

[44] I. Kevin, Kai Wang, Xiaokang Zhou, Wei Liang, Zheng Yan, and Jinhua She. 2021. Federated transfer learning based cross-domain prediction for smart manufacturing. *IEEE Transactions on Industrial Informatics* 18, 6 (2021), 4088–4096.

[45] Goran D. Putnik, Vijaya Kumar Manupati, Sai Krishna Pabba, Leonilde Varela, and Francisco Ferreira. 2021. Semi-double-loop machine learning based CPS approach for predictive maintenance in manufacturing system based on machine status indications. *CIRP Annals* 70, 1 (2021), 365–368.

[46] Seunghyeon Kim, Yuchang Won, Kyung-Joon Park, and Yongsoon Eun. 2022. A data-driven indirect estimation of machine parameters for smart production systems. *IEEE Transactions on Industrial Informatics* 18, 10 (2022), 6537–6546.

[47] Zhenghong Wu, Hongkai Jiang, Tengfei Lu, and Ke Zhao. 2020. A deep transfer maximum classifier discrepancy method for rolling bearing fault diagnosis under few labeled data. *Knowledge-Based Systems* 196 (2020), 105814.

[48] Yi Qin, Qunwang Yao, Yi Wang, and Yongfang Mao. 2021. Parameter sharing adversarial domain adaptation networks for fault transfer diagnosis of planetary gearboxes. *Mechanical Systems and Signal Processing* 160 (2021), 107936.

[49] Amin Shahraki, Mahmoud Abbasi, Amir Taherkordi, and Anca Delia Jurcut. 2021. Active learning for network traffic classification: A technical study. *IEEE Transactions on Cognitive Communications and Networking* 8, 1 (2021), 422–439.

[50] Jaewoong Shim, Seokho Kang, and Sungzoon Cho. 2020. Active learning of convolutional neural network for cost-effective wafer map pattern classification. *IEEE Transactions on Semiconductor Manufacturing* 33, 2 (2020), 258–266.

[51] Tamara Todic, Vladimir Stankovic, and Lina Stankovic. 2023. An active learning framework for the low-frequency non-intrusive load monitoring problem. *Applied Energy* 341 (2023), 121078.

[52] Kanhong Xiao, Jiangzhong Cao, Zekai Zeng, and Wing-Kuen Ling. 2023. Graph-based active learning with uncertainty and representativeness for industrial anomaly detection. *IEEE Transactions on Instrumentation and Measurement* 72 (2023), 1–14.

[53] Jesper E. Van Engelen and Holger H. Hoos. 2020. A survey on semi-supervised learning. *Machine Learning* 109, 2 (2020), 373–440.

[54] Shamsul Huda, Suruz Miah, Mohammad Mehedi Hassan, Rafiqul Islam, John Yearwood, Majed Alrubaian, and Ahmad Almogren. 2017. Defending unknown attacks on cyber-physical systems by semi-supervised approach and available unlabeled data. *Information Sciences* 379 (2017), 211–228.

[55] Guoteng Wang, Chongyu Wang, Mohammad Shahidehpour, and Wei Lin. 2023. Deep semi-supervised learning method for false data detection against forgery and concealing of faults in cyber-physical power systems. *IEEE Transactions on Smart Grid* 15, 1 (2023), 944–958.

[56] João Gama, Indrė Žliobaitė, Albert Bifet, Mykola Pechenizkiy, and Abdelhamid Bouchachia. 2014. A survey on concept drift adaptation. *ACM Computing Surveys* 46, 4 (2014), 1–37.

[57] Günter Bitsch and Felicia Schweitzer. 2022. Selection of optimal machine learning algorithm for autonomous guided vehicle's control in a smart manufacturing environment. *Procedia CIRP* 107 (2022), 1409–1414.

[58] Gaoqing Shen, Lei Lei, Zhilin Li, Shengsuo Cai, Lijuan Zhang, Pan Cao, and Xiaojiao Liu. 2021. Deep reinforcement learning for flocking motion of multi-UAV systems: Learn from a digital twin. *IEEE Internet of Things Journal* 9, 13 (2021), 11141–11153.

[59] Roberto Bodo, Matteo Bertocco, and Alberto Bianchi. 2020. Feature ranking under industrial constraints in continuous monitoring applications based on machine learning techniques. In *Proceeding of the 2020 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*. IEEE, 1–6.

[60] Chun-Cheng Lin, Der-Jiunn Deng, Chin-Hung Kuo, and Linnan Chen. 2019. Concept drift detection and adaption in big imbalance industrial IoT data using an ensemble learning method of offline classifiers. *IEEE Access* 7 (2019), 56198–56207.

[61] Vishalteja Kosana, Madasthu Santhosh, Kiran Teeparthi, and Santosh Kumar. 2022. A novel dynamic selection approach using on-policy SARSA algorithm for accurate wind speed prediction. *Electric Power Systems Research* 212 (2022), 108174.

[62] Xianglong Zhou, Yun Lin, Ya Tu, Shiwen Mao, and Zheng Dou. 2019. Dynamic channel allocation for multi-UAVs: A deep reinforcement learning approach. In *Proceedings of the 2019 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 1–6.

[63] Dinithi Jayaratne, Daswin De Silva, Damminda Alahakoon, and Xinghuo Yu. 2021. Continuous detection of concept drift in industrial cyber-physical systems using closed loop incremental machine learning. *Discover Artificial Intelligence* 1 (2021), 1–13.

[64] Hind Bangui, Mouzhi Ge, and Barbora Buhnova. 2022. A hybrid machine learning model for intrusion detection in VANET. *Computing* 104, 3 (2022), 503–531.

[65] HongLei Che. 2021. Multi-sensor data fusion method based on ARIMA-LightGBM for AGV positioning. In *Proceedings of the 2021 5th International Conference on Robotics and Automation Sciences (ICRAS)*. IEEE, 272–276.

[66] Miseul Kim, Minh Tri Ho, and Hong-Goo Kang. 2021. Self-supervised complex network for machine sound anomaly detection. In *Proceedings of the 2021 29th European Signal Processing Conference (EUSIPCO)*. IEEE, 586–590.

[67] Qianqian Zhou, Nan Chen, and Siwei Lin. 2022. FASTNN: A deep learning approach for traffic flow prediction considering spatiotemporal features. *Sensors* 22, 18 (2022).

[68] Sihoon Moon, Sanghoon Lee, and Kyung-Joon Park. 2023. Graph-based reinforcement learning for flexible job shop scheduling with transportation constraints. In *Proceedings of the 49th Annual Conference of the IEEE Industrial Electronics Society (IECON)*, 1–6.

[69] Qinghua Xu, Shaukat Ali, and Tao Yue. 2021. Digital twin-based anomaly detection in cyber-physical systems. In *Proceedings of the 2021 14th IEEE Conference on Software Testing, Verification and Validation (ICST)*. IEEE, 205–216.

[70] Florian Bachinger, Gabriel Kronberger, and Michael Affenzeller. 2021. Continuous improvement and adaptation of predictive models in smart manufacturing and model management. *IET Collaborative Intelligent Manufacturing* 3, 1 (2021), 48–63.

[71] Benjamin Maschler, Timo Müller, Andreas Löcklin, and Michael Weyrich. 2022. Transfer learning as an enhancement for reconfiguration management of cyber-physical production systems. *Procedia CIRP* 112 (2022), 220–225.

[72] Linghe Kong, Mingyuan Xia, Xiao-Yang Liu, Guangshuo Chen, Yu Gu, Min-You Wu, and Xue Liu. 2013. Data loss and reconstruction in wireless sensor networks. *IEEE Transactions on Parallel and Distributed Systems* 25, 11 (2013), 2818–2828.

[73] José M. Jerez, Ignacio Molina, Pedro J. García-Laencina, Emilio Alba, Nuria Ribelles, Miguel Martín, and Leonardo Franco. 2010. Missing data imputation using statistical and machine learning methods in a real breast cancer problem. *Artificial Intelligence in Medicine* 50, 2 (2010), 105–115.

[74] Rahmat Nur Faizin, Mardhani Riasetiawan, and Ahmad Ashari. 2019. A review of missing sensor data imputation methods. In *Proceedings of the 2019 5th International Conference on Science and Technology (ICST), Vol.* 1. IEEE, 1–6.

[75] Yujue Zhou, Jie Jiang, Shuang-Hua Yang, Ligang He, and Yulong Ding. 2021. MuSDRI: Multi-seasonal decomposition based recurrent imputation for time series. *IEEE Sensors Journal* 21, 20 (2021), 23213–23223.
[76] Soohwan Jeong, Chonghyo Joo, Jongkoo Lim, Hyungtae Cho, Sungsu Lim, and Junghwan Kim. 2023. A novel graph-based missing values imputation method for industrial lubricant data. *Computers in Industry* 150 (2023), 103937.
[77] Marco Ramoni and Paola Sebastiani. 2001. Robust learning with missing data. *Machine Learning* 45 (2001), 147–170.
[78] Zhengdao Zhang and Feilong Dong. 2014. Fault detection and diagnosis for missing data systems with a three time-slice dynamic Bayesian network approach. *Chemometrics and Intelligent Laboratory Systems* 138 (2014), 30–40.
[79] M. Premkumar, R. Lakshmi, P. Velrajkumar, S. Gayathri Priya, Rama Chaithanya Tanguturi, S. Murali, and M. Sivaramkrishnan. 2023. Hybrid deep learning model for cyber-attack detection. In *Proceedings of the 2023 7th International Conference on Intelligent Computing and Control Systems (ICICCS)*. IEEE, 1435–1441.
[80] Nikhil Muralidhar, Sathappan Muthiah, Kiyoshi Nakayama, Ratnesh Sharma, and Naren Ramakrishnan. 2019. Multivariate long-term state forecasting in cyber-physical systems: A sequence to sequence approach. In *Proceedings of the 2019 IEEE International Conference on Big Data (Big Data)*, 543–552.
[81] Jingyu Yang, Zuogong Yue, and Ye Yuan. 2023. Deep probabilistic graphical modeling for robust multivariate time series anomaly detection with missing data. *Reliability Engineering & System Safety* 238 (2023), 109410.
[82] O. Llanes-Santiago, B. C. Rivero-Benedico, S. C. Gálvez-Viera, E. F. Rodríguez-Morant, R. Torres-Cabeza, and A. J. Silva-Neto. 2019. A fault diagnosis proposal with online imputation to incomplete observations in industrial plants. *Revista Mexicana de Ingeniería Química* 18, 1 (2019), 83–98.
[83] Kisan Sarda, Amol Yerudkar, and Carmen Del Vecchio. 2021. Missing data imputation for real time-series data in a steel industry using generative adversarial networks. In *Proceedings of the 47th Annual Conference of the IEEE Industrial Electronics Society (IECON). IEEE*, 1–6.
[84] C. Warren Axelrod. 2013. Managing the risks of cyber-physical systems. In *Proceedings of the 2013 IEEE Long Island Systems, Applications and Technology Conference (LISAT)*. IEEE, 1–6.
[85] Xiaoge Zhang, Felix T. S. Chan, Chao Yan, and Indranil Bose. 2022. Towards risk-aware artificial intelligence and machine learning systems: An overview. *Decision Support Systems* 159 (2022), 113800.
[86] Sebastian Jaimungal, Silvana M. Pesenti, Ye Sheng Wang, and Hariom Tatsat. 2022. Robust risk-aware reinforcement learning. *SIAM Journal on Financial Mathematics* 13, 1 (2022), 213–226.
[87] Mohammad Sayad Haghighi, Faezeh Farivar, and Alireza Jolfaei. 2020. A machine-learning-based approach to build zero-false-positive IPSs for industrial IoT and CPS with a case study on power grids security. *IEEE Transactions on Industry Applications* 60, 1 (2020), 920–928.
[88] Xiaozhe Gu and Arvind Easwaran. 2019. Towards safe machine learning for CPS: Infer uncertainty from training data. In *Proceedings of the 10th ACM/IEEE International Conference on Cyber-Physical Systems (ICCPS '19)*. ACM, New York, NY, 249–258.
[89] Xugui Zhou, Bulbul Ahmed, James H. Aylor, Philip Asare, and Homa Alemzadeh. 2024. Hybrid knowledge and data driven synthesis of runtime monitors for cyber-physical systems. *IEEE Transactions on Dependable and Secure Computing* 21, 1 (2024), 12–30.
[90] Hansong Xu, Xing Liu, Wei Yu, David Griffith, and Nada Golmie. 2020. Reinforcement learning-based control and networking co-design for industrial Internet of things. *IEEE Journal on Selected Areas in Communications* 38, 5 (2020), 885–898.
[91] Hyung-Seok Park, Sihoon Moon, Jeongho Kwak, and Kyung-Joon Park. 2022. CAPL: Criticality-aware adaptive path learning for industrial wireless sensor-actuator networks. *IEEE Transactions on Industrial Informatics* 19, 8 (2022), 9123–9133.
[92] Deyun Lyu, Jiayang Song, Zhenya Zhang, Zhijie Wang, Tianyi Zhang, Lei Ma, and Jianjun Zhao. 2023. Autorepair: Automated repair for AI-enabled cyber-physical systems under safety-critical conditions. arXiv:2304.05617. Retrieved from https://doi.org/10.48550/arXiv.2304.05617
[93] Ferhat Ozgur Catak, Tao Yue, and Shaukat Ali. 2022. Uncertainty-aware prediction validator in deep learning models for cyber-physical system data. *ACM Transactions on Software Engineering and Methodology* 31, 4 (2022), 1–31.
[94] Jiayang Song, Deyun Lyu, Zhenya Zhang, Zhijie Wang, Tianyi Zhang, and Lei Ma. 2022. When cyber-physical systems meet AI: A benchmark, an evaluation, and a way forward. In *Proceedings of the 44th International Conference on Software Engineering: Software Engineering in Practice (ICSE)*, 343–352.
[95] Ga-Yeong Kim, Su-Min Lim, and Ieck-Chae Euom. 2022. A study on performance metrics for anomaly detection based on industrial control system operation data. *Electronics* 11, 8 (2022), 1213.
[96] Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador García, Sergio Gil-López, Daniel Molina, Richard Benjamins, et al. 2020. Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion* 58 (2020), 82–115.

[97] Zakaria Abou El Houda, Bouziane Brik, and Lyes Khoukhi. 2022. "Why should I trust your IDs?": An explainable deep learning framework for intrusion detection systems in Internet of things networks. *IEEE Open Journal of the Communications Society* 3 (2022), 1164–1176.

[98] S. Sivamohan, S. S. Sridhar, and S. Krishnaveni. 2023. TEA-EKHO-IDS: An intrusion detection system for industrial CPS with trustworthy explainable AI and enhanced krill herd optimization. *Peer-to-Peer Networking and Applications* 16, 4 (2023), 1993–2021.

[99] Ioannis T. Christou, Nikos Kefalakis, Andreas Zalonis, and John Soldatos. 2020. Predictive and explainable machine learning for industrial Internet of things applications. In *Proceedings of the 2020 16th International Conference on Distributed Computing in Sensor Systems (DCOSS)*. IEEE, 213–218.

[100] Manikandasriram Srinivasan Ramanagopal, Cyrus Anderson, Ram Vasudevan, and Matthew Johnson-Roberson. 2018. Failing to learn: Autonomously identifying perception failures for self-driving cars. *IEEE Robotics and Automation Letters* 3, 4 (2018), 3860–3867.

[101] Dimitrios Boursinos and Xenofon Koutsoukos. 2021. Assurance monitoring of learning-enabled cyber-physical systems using inductive conformal prediction based on distance learning. *AI EDAM* 35, 2 (2021), 251–264.

[102] Demetris Trihinas, Michalis Agathocleous, and Karlen Avogian. 2021. Composable energy modeling for ML-driven drone applications. In *Proceedings of the 2021 IEEE International Conference on Cloud Engineering (IC2E)*. IEEE, 231–237.

[103] Marwan Dhuheir, Emna Baccour, Aiman Erbad, Sinan Sabeeh, and Mounir Hamdi. 2021. Efficient real-time image recognition using collaborative swarm of UAVs and convolutional networks. In *Proceedings of the 2021 International Wireless Communications and Mobile Computing (IWCMC)*. IEEE, 1954–1959.

[104] Rockson Agyeman and Bernhard Rinner. 2022. Resource-efficient pervasive smart camera networks. In *Proceedings of the 2022 IEEE International Conference on Pervasive Computing and Communications Workshops and Other Affiliated Events (PerCom Workshops)*. IEEE, 503–508.

[105] Lei Lei, Gaoqing Shen, Lijuan Zhang, and Zhilin Li. 2020. Toward intelligent cooperation of UAV swarms: When machine learning meets digital twin. *IEEE Network* 35, 1 (2020), 386–392.

[106] Xin Tang, Xiaohuan Li, Rong Yu, Yuan Wu, Jin Ye, Fengzhu Tang, and Qian Chen. 2023. Digital-twin-assisted task assignment in multi-UAV systems: A deep reinforcement learning approach. *IEEE Internet of Things Journal* 10, 17 (2023), 15362–15375.

[107] Georgios Tertytchny and Maria K. Michael. 2020. Dataset reduction framework for intelligent fault detection in IoT-based cyber-physical systems using machine learning techniques. In *Proceedings of the 2020 International Conference on Omni-Layer Intelligent Systems (COINS)*. IEEE, 1–6.

[108] Onder Tutsoy, Duygun Erol Barkana, and Sule Colak. 2017. Learning to balance an NAO robot using reinforcement learning with symbolic inverse kinematic. *Transactions of the Institute of Measurement and Control* 39, 11 (2017), 1735–1748.

[109] Mohammadhossein Ghahramani, Yan Qiao, Meng Chu Zhou, Adrian O'Hagan, and James Sweeney. 2020. AI-based modeling and data-driven evaluation for smart manufacturing processes. *IEEE/CAA Journal of Automatica Sinica* 7, 4 (2020), 1026–1037.

[110] Jie Zhong, Yujie Zhang, Jianyu Wang, Chong Luo, and Qiang Miao. 2021. Unmanned aerial vehicle flight data anomaly detection and recovery prediction based on spatio-temporal correlation. *IEEE Transactions on Reliability* 71, 1 (2021), 457–468.

[111] Romany F. Mansour. 2022. Artificial intelligence based optimization with deep learning model for blockchain enabled intrusion detection in CPS environment. *Scientific Reports* 12, 1 (2022), 12937.

[112] Martin Kiac, Pavel Sikora, Lukas Malina, Zdenek Martinasek, and Gautam Srivastava. 2023. ADEROS: Artificial intelligence-based detection system of critical events for road security. *IEEE Systems Journal* 17, 4 (2023), 5073–5084.

[113] Yilan Li, Haowen Fang, Mingyang Li, Yue Ma, and Qinru Qiu. 2022. Neural network pruning and fast training for DRL-based UAV trajectory planning. In *Proceedings of the 2022 27th Asia and South Pacific Design Automation Conference (ASP-DAC)*, 574–579.

[114] Lei Ren, Tao Wang, Zidi Jia, Fangyu Li, and Honggui Han. 2022. A lightweight and adaptive knowledge distillation framework for remaining useful life prediction. *IEEE Transactions on Industrial Informatics* 19, 8 (2022), 9060–9070.

[115] Yi Jin, Bin Huang, Yulong Yan, Yuxiang Huan, Jiawei Xu, Shancang Li, Prosanta Gope, Li Da Xu, Zhuo Zou, and Lirong Zheng. 2022. Edge-based collaborative training system for artificial intelligence-of-things. *IEEE Transactions on Industrial Informatics* 18, 10 (2022), 7162–7173.

[116] Jed Mills, Jia Hu, and Geyong Min. 2019. Communication-efficient federated learning for wireless edge intelligence in IoT. *IEEE Internet of Things Journal* 7, 7 (2019), 5986–5994.

[117] Shui Jiang, Xiaoding Wang, Youxiong Que, and Hui Lin. 2024. Fed-MPS: Federated learning with local differential privacy using model parameter selection for resource-constrained CPS. *Journal of Systems Architecture* 150 (2024), 103108.

[118] Pietro Cassará, Alberto Gotta, and Lorenzo Valerio. 2022. Federated feature selection for cyber-physical systems of systems. *IEEE Transactions on Vehicular Technology* 71, 9 (2022), 9937–9950.

[119] Suwani Jayasinghe, Yushan Siriwardhana, Pawani Porambage, Madhusanka Liyanage, and Mika Ylianttila. 2022. Federated learning based anomaly detection as an enabler for securing network and service management automation in beyond 5G networks. In *Proceedings of the 2022 Joint European Conference on Networks and Communications & 6G Summit (EuCNC/6G Summit)*. IEEE, 345–350.

[120] Viraaji Mothukuri, Prachi Khare, Reza M. Parizi, Seyedamin Pouriyeh, Ali Dehghantanha, and Gautam Srivastava. 2021. Federated-learning-based anomaly detection for IoT security attacks. *IEEE Internet of Things Journal* 9, 4 (2021), 2545–2554.

[121] Beibei Li, Yuhao Wu, Jiarui Song, Rongxing Lu, Tao Li, and Liang Zhao. 2021. DeepFed: Federated deep learning for intrusion detection in industrial cyber–physical systems. *IEEE Transactions on Industrial Informatics* 17, 8 (2021), 5615–5624.

[122] Ahmad Zainudin, Rubina Akter, Dong-Seong Kim, and Jae-Min Lee. 2023. Federated learning inspired low-complexity intrusion detection and classification technique for SDN-based industrial CPS. *IEEE Transactions on Network and Service Management* 20, 3 (2023), 2442–2459.

[123] Seongjin Yun, Deun-Sol Cho, Hyeong-su Kim, Hanjin Kim, and Won-Tae Kim. 2021. An intelligence-defined networking architecture with importance-based network resource control. *IEEE Internet of Things Journal* 10, 4 (2021), 2922–2933.

[124] Wei Yang Bryan Lim, Sahil Garg, Zehui Xiong, Yang Zhang, Dusit Niyato, Cyril Leung, and Chunyan Miao. 2021. UAV-assisted communication efficient federated learning in the era of the artificial intelligence of things. *IEEE Network* 35, 5 (2021), 188–195.

[125] Zhaoyan Shen, Qingxiang Tang, Tianren Zhou, Yuhao Zhang, Zhiping Jia, Dongxiao Yu, Zhiyong Zhang, and Bingzhe Li. 2023. Ashl: An adaptive multi-stage distributed deep learning training scheme for heterogeneous environments. *IEEE Transactions on Computers* 73, 1 (2023), 30–43.

[126] Haitao Zhao, Yuhao Tan, Kun Guo, Wenchao Xia, Bo Xu, and Tony Q. S. Quek. 2024. Client scheduling for multiserver federated learning in industrial IoT with unreliable communications. *IEEE Internet of Things Journal* 11, 9 (2024), 16478–16490.

[127] Wen Sun, Jiajia Liu, and Yanlin Yue. 2019. AI-enhanced offloading in edge computing: When machine learning meets industrial IoT. *IEEE Network* 33, 5 (2019), 68–74.

[128] Xiaomin Li, Jiafu Wan, Hong-Ning Dai, Muhammad Imran, Min Xia, and Antonio Celesti. 2019. A hybrid computing solution and resource scheduling strategy for edge computing in smart manufacturing. *IEEE Transactions on Industrial Informatics* 15, 7 (2019), 4225–4234.

[129] Vivian Ukamaka Ihekoronye, Simeon Okechukwu Ajakwe, Dong-Seong Kim, and Jae Min Lee. 2022. Cyber edge intelligent intrusion detection framework for UAV network based on random forest algorithm. In *Proceedings of the 2022 13th International Conference on Information and Communication Technology Convergence (ICTC)*. IEEE, 1242–1247.

[130] Sha Zhu, Kaoru Ota, and Mianxiong Dong. 2021. Green AI for IIoT: Energy efficient intelligent edge computing for industrial Internet of things. *IEEE Transactions on Green Communications and Networking* 6, 1 (2021), 79–88.

[131] Latif U. Khan, Madyan Alsenwi, Ibrar Yaqoob, Muhammad Imran, Zhu Han, and Choong Seon Hong. 2020. Resource optimized federated learning-enabled cognitive Internet of things for smart industries. *IEEE Access* 8 (2020), 168854–168864.

[132] Zhijian He, Yao Chen, Enyan Huang, Qixin Wang, Yu Pei, and Haidong Yuan. 2019. A system identification based oracle for control-CPS software fault localization. In *Proceedings of the 2019 IEEE/ACM 41st International Conference on Software Engineering (ICSE)*. IEEE, 116–127.

[133] Afsoon Afzal, Claire Le Goues, and Christopher Steven Timperley. 2021. Mithra: Anomaly detection as an oracle for cyberphysical systems. *IEEE Transactions on Software Engineering* 48, 11 (2021), 4535–4552.

[134] Chih-Ling Chang, Jui-Lung Hung, Chin-Wei Tien, Chia-Wei Tien, and Sy-Yen Kuo. 2020. Evaluating robustness of AI models against adversarial attacks. In *Proceedings of the 1st ACM Workshop on Security and Privacy on Artificial Intelligence (ASIA CCS)*, 47–54.

[135] Jiangnan Li, Yingyuan Yang, Jinyuan Stella Sun, Kevin Tomsovic, and Hairong Qi. 2021. ConAML: Constrained adversarial machine learning for cyber-physical systems. In *Proceedings of the 2021 ACM Asia Conference on Computer and Communications Security (ASIACCS)*, 52–66.

[136] Anay Pattanaik, Zhenyi Tang, Shuijing Liu, Gautham Bommannan, and Girish Chowdhary. 2018. Robust deep reinforcement learning with adversarial attacks. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS '18)*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 2040–2042.

[137] Kexin Pei, Yinzhi Cao, Junfeng Yang, and Suman Jana. 2017. Deepxplore: Automated whitebox testing of deep learning systems. *In Proceedings of the 26th Symposium on Operating Systems Principles (SOSP)*, 1–18.

[138] Xugui Zhou, Maxfield Kouzel, and Homa Alemzadeh. 2022. Robustness testing of data and knowledge driven anomaly detection in cyber-physical systems. In *Proceedings of the 2022 52nd Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W)*. IEEE, 44–51.

[139] Yuntao Wang, Zhou Su, Ning Zhang, and Abderrahim Benslimane. 2020. Learning in the air: Secure federated learning for UAV-assisted crowdsensing. *IEEE Transactions on Network Science and Engineering* 8, 2 (2020), 1055–1069.

[140] Lei Cui, Youyang Qu, Gang Xie, Deze Zeng, Ruidong Li, Shigen Shen, and Shui Yu. 2021. Security and privacy-enhanced federated learning for anomaly detection in IoT infrastructures. *IEEE Transactions on Industrial Informatics* 18, 5 (2021), 3492–3500.

[141] Yuzheng Ren, Renchao Xie, Fei Richard Yu, Tao Huang, and Yunjie Liu. 2022. Green intelligence networking for connected and autonomous vehicles in smart cities. *IEEE Transactions on Green Communications and Networking* 6, 3 (2022), 1591–1603.

[142] Sushil Kumar Singh, Shailendra Rathore, and Jong Hyuk Park. 2020. Blockiotintelligence: A blockchain-enabled intelligent IoT architecture with artificial intelligence. *Future Generation Computer Systems* 110 (2020), 721–743.

Received 30 July 2024; revised 23 December 2024; accepted 21 February 2025