Article | Received 8 October 2024; Accepted 16 December 2024; Published 7 January 2025
https://doi.org/10.55092/am20250002

Recovery of cyber manufacturing systems from false data injection attacks on sensors using reinforcement learning

Romesh Prasad and Young Moon*

Department of Mechanical and Aerospace Engineering, Syracuse University, New York, USA

* Correspondence author; E-mail: ybmoon@syr.edu.

Highlights:

- Cyber-attacks against cyber-manufacturing systems are new class of probelms.
- False data injection attack on sensors is one of the critical problems.
- An innovative approach to address the problem using reinforcement learning is proposed.

Abstract: The integration of automation, connectivity, and advanced analytics in manufacturing enhances productivity but also increases vulnerability to cyber threats including sensor attacks. Sensors—critical to automation—are particularly susceptible to false data injection attacks that can disrupt operations and lead to system failures. Despite advancements in prevention and detection methods, effective post-attack recovery remains an underexplored area—critical to minimizing operational downtime in manufacturing. The research addresses this gap by introducing a novel agent-based recovery modeling approach tailored for manufacturing systems. A reinforcement learning-driven recovery strategy is developed to restore operations efficiently after sensor attacks. The approach is evaluated through two distinct sensor attack scenarios. The recovery agent's performance is benchmarked against a PID controller using key metrics: downtime, throughput, and efficiency. Results demonstrate significant improvements by enhancing the resilience and security of manufacturing systems against sensor attacks.

Keywords: manufacturing system; deep reinforcement learning; sensor attacks; resiliency; recovery

1. Introduction

Cyber manufacturing systems (CMS) are designed to adapt to achieve the intended objectives by the individual manufacturing processes, their relationship, and interactions with the cyber components and the operational environment . The interaction between and among cyber components and manufacturing systems is achieved through wired or wireless networks [1]. However, this integration and interplay between traditional manufacturing and cyber elements also expose these systems to potential threats from cyber attackers [1]. According to International Business Machines (IBM) [2], manufacturing industries ranked second in 2021 and it claimed the top spot in 2022 for the most attacked operational

technology (OT) industry. Furthermore, some of these attacks after infiltration targeted the manufacturing controller, halting the manufacturing systems and increasing the downtime. Also, given the low tolerance of manufacturers in downtime, demanding ransom in response to the removal of the malware seemed a lucrative strategy for attackers.

Among different security risks to cyber manufacturing systems, sensor attacks stand out as a threat because of their potential to significantly disrupt manufacturing processes and systems. Automation in manufacturing can be achieved using a framework such as the five-layer industrial automation framework as depicted in Figure 1. Sensors are instrumental in the communication of information from process levels to enterprise levels within such a framework [3]. Sensors serve as the pivotal part of this framework, enabling controllers at the process level and operational or supervisory level to implement precise control algorithms that enable the functioning of cyber manufacturing systems in a predetermined sequence to yield the desired end product [4]. Moreover, they provide real-time state estimation of the manufacturing systems to the manufacturing plant level controller. With this information, the manufacturing plant-level control algorithms work to minimize any discrepancies between the manufacturing system's output and the input required for optimal manufacturing operation.

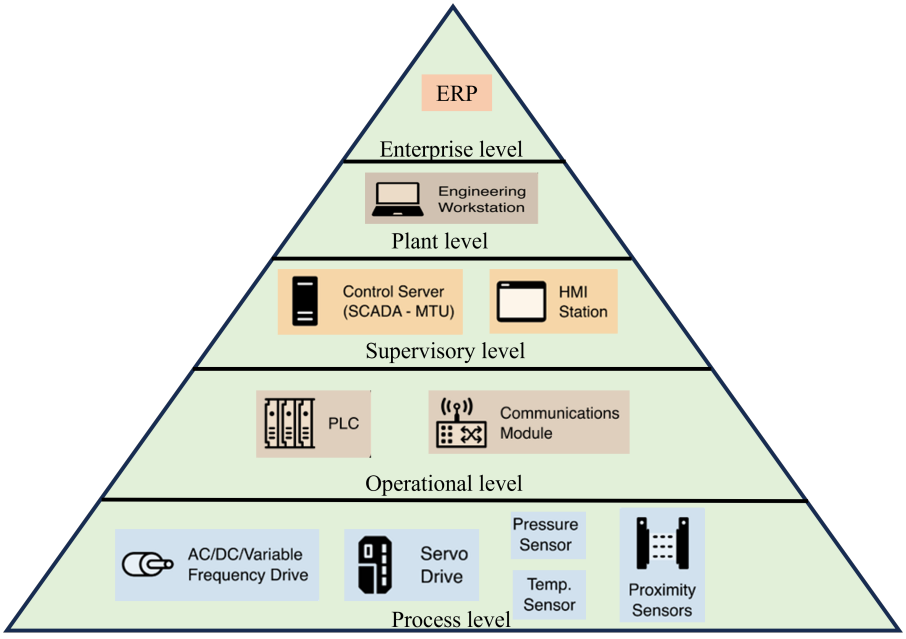

Figure 1. Manufacturing systems automation hierarchy.

At the highest level of the automation framework, the Manufacturing Execution System (MES) and Enterprise Resource Planning (ERP) integrate production planning, scheduling, and resource allocation, bridging the gap between supervisory control and business planning. However, the integrity of the information can be compromised by the actions of cyber attackers who may exploit vulnerabilities in the system's networks or software, hence jeopardizing the manufacturing system's availability and potentially leading it into an unsafe operational state.

Cyber manufacturing systems face a variety of cyber threats, prompting researchers to develop solutions aimed at increasing resiliency. Broadly, these solutions can be categorized into prevention and detection strategies. Prevention strategies enforce constraints through rules, policies, or technologies such

as firewalls, physical hash functions, and blockchain [5–7]. While these measures are crucial, they cannot eliminate all possible cyber threats; making detection strategies essential. Detection focuses on identifying and flagging anomalous behavior within the system, with approaches tailored to different manufacturing scenarios extensively covered in literature [8–12]. However, a critical gap lies in post-detection response and recovery.

The existing literature asserts that complete resilience for any physical system can only be achieved when a system optimally recovers from a cyber-attack [13]. Current recovery strategies—categorized into forward rolling and backward rolling—offer insights from general physical systems [14]. Backward rolling, which restores a system to its last safe state, is impractical for many cyber manufacturing systems due to their irreversible processes. Forward rolling, which projects the system to a future safe state, offers a more viable alternative. Yet, applying this approach to CMS is challenging due to the complexity of their multi-level automation frameworks and their discrete stochastic nonlinear behavior [15]. This paper addresses this critical challenge by tailoring forward recovery to enhance the resiliency of CMS against sensor attacks.

Considering the challenges and limitations highlighted above, this research aims to investigate how to respond effectively once a sensor attack is detected from a cyber manufacturing systems perspective. This research aims to provide insights into developing recovery strategies that can (i) restore the impacted manufacturing system from a sensor attack, and (ii) minimize the downtime experienced by cyber manufacturing systems due to the sensor attack. To achieve the objectives, this research:

(1) conducted a comprehensive exploration of the challenges and complexities associated with developing recovery strategies for cyber manufacturing systems, and
(2) developed a recovery agent based on reinforcement learning. This reinforcement agent plays a pivotal role in recovering the cyber manufacturing systems and minimizes downtime to enhance the overall resilience of cyber manufacturing systems.

The paper is structured as follows. Section 2 discusses the background, highlights the challenges associated with developing recovery strategies for cyber manufacturing systems, and formulates the problem. Section 3 presents the manufacturing systems overview, threat model, and the proposed recovery strategy. Section 4 introduces reinforcement learning while Section 5 focuses on presenting the results obtained and engaging in a discussion of the findings in the context of recovery strategies for cyber manufacturing systems.

2. Background and problem formulation

In this section, the current status of security research for cyber manufacturing systems, the gap in the research, and the challenges of developing a successful recovery strategy are presented.

2.1. Research in CMS security

One prominent case illustrating the potential consequences of cyber-attacks is the infamous Stuxnet incident [16]. Stuxnet demonstrated the ability of malicious code to take control of sensors and manufacturing system controllers once it infiltrates the target systems, resulting in severe operational disruptions. Another variant of these sensor attacks that are well studied from the context of industrial

control systems is false data injection attacks on sensors. In today's highly automated and interconnected manufacturing environments, sensors serve as the eyes and ears of the system, providing real-time data, critical for process control and decision-making. False data injection attacks, which involve tampering with sensor data to deceive the system, can have a severe impact. They can lead to defective products, equipment malfunctions, and even compromise worker safety. Researchers are drawn to address this issue through development of prevention and detection strategies.

2.2. Research gap and scope

While much research has focused on the prevention and detection of sensor attacks in cyber manufacturing systems, there is a noticeable gap in understanding what to do after an attack is detected. This gap is addressed in this research through a reactive recovery strategy triggered by a detection algorithm. However, since the accuracy of the recovery strategy is dependent on the accuracy of the detection, precise detection systems are much needed.

2.3. Researchers addressing recovery

Given the limited prior research on the scope of this paper, our exploration of existing literature delves into the recovery of physical systems from sensor attacks. In this context, Kong highlights the necessity of a backup controller algorithm to facilitate the recovery of a physical process after sensor attacks occur [14]. The literature on recovery can be categorized into two strategies: forward rolling and backward rolling. In forward rolling, the future system state is estimated and the system is moved into the projected future state. The backward rolling is the opposite, it tracks the last known safe state and rolls the system to that state. According to Kong, this approach is burdened by overhead costs and is often rendered unfeasible due to the irreversible nature of many manufacturing processes.

Consequently, the focus has predominantly been on forward recovery, which appears to be a more practical and viable strategy. In the context of forward recovery, achieving a precise estimation of the system's future state is challenging [14]. This estimation can be accomplished by modeling the systems as linear time-invariant and implementing various control policies such as linear quadratic regulators [17], linear approximations [17], and predictive control [18]. These control strategies estimate the future state and roll the system into future states. An alternative approach to forecasting the future state of the system is to employ data-driven modeling techniques. This entails utilizing deep learning methods like long short-term memory (LSTM) to predict the system's future state based on historical data [19]. Such data-driven models can enhance the accuracy of forward recovery strategies, making them more effective in mitigating the impacts of sensor attacks on manufacturing processes.

2.4. Challenges of developing a recovery strategy for cyber manufacturing systems

Adapting the forward-rolling approach for cyber manufacturing systems presents a significant challenge due to the distinctive automation architectures in place. These manufacturing systems utilize a certain system architecture such as a five-level automation hierarchy depicted in Figure 1. Information from sensors is exchanged across and within these hierarchy levels, making it essential to consider all levels when implementing the forward rolling recovery strategy. Only a process-level recovery is not sufficient

to successfully recover a manufacturing system. Communication between all levels at the same time is desired. Hence a recovery strategy should consider multiple manufacturing processes, their sequences, and production planning. In contrast, existing literature solutions typically concentrate on individual processes such as DC motor control, quad-copter navigation, or vehicle speed regulation [14, 17–20]. Furthermore, modeling cyber manufacturing systems for recovery proves to be a challenge, given their discrete, stochastic, and nonlinear nature—making it challenging to establish closed form solutions or precise mathematical representations. This stark contrast in scale, complexity, and modeling challenges underscores the difficulties of adapting recovery strategies designed for cyber-physical systems to the intricate and multifaceted world of cyber manufacturing systems.

2.5. Recovery from cyber manufacturing system perspective

An ideal recovery strategy for a cyber manufacturing system should go beyond restoring the impacted manufacturing system's state and extend to ensuring continuous, uninterrupted operation. A conceptual recovery model for a cyber manufacturing system is depicted in Figure 2. The x-axis in this figure represents different stages of a manufacturing system, such as warm-up time, cycle time, and cooling time. Under normal operating conditions, predefined processes are executed through controllers. However, in the event of a false data injection attack on sensors, these controllers can lead the system to undesirable states. One straightforward approach to achieve recovery in such a situation is to shut down the process and restart it. However, as depicted in the Figure 2, the warm-up time consumes a significant amount of production time, making this option less feasible. An alternative approach involves implementing the forward-rolling approach, where the impacted manufacturing systems are restored to their original operating conditions. This can be achieved by developing alternate control algorithms. However, merely returning it to a normal operating state is insufficient. In an automation hierarchy, any disruption in the discrete events can cause other subprocesses to start and stop abruptly. Therefore, the recovery process must ensure that the manufacturing system not only returns to a normal operating condition but also continues to operate seamlessly. This continuous operation is achieved through recovery strategies in operational and supervisory controllers. By implementing a comprehensive recovery strategy that encompasses all these elements, a cyber manufacturing system can recover from disruptions and maintain its operational efficiency while minimizing production losses and downtime.

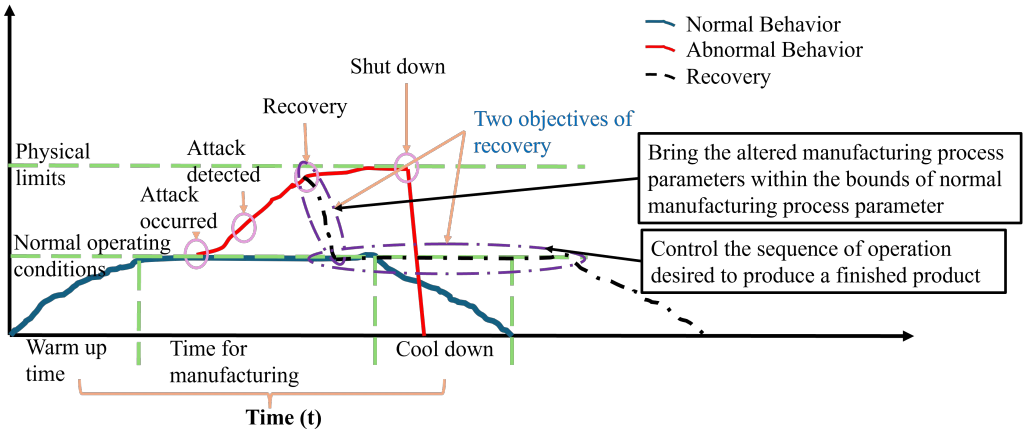

Figure 2. Conceptual -Manufacturing systems recovery.

2.6. Problem formulation

The problem under investigation in this study pertains to the recovery of cyber manufacturing systems, which poses distinct challenges compared to the recovery of both pure cyber systems and cyber-physical systems. Unlike typical recovery efforts in cyber systems, which primarily focus on computational or data recovery, and recovery in cyber-physical systems, which are geared towards physical systems recovery, the recovery challenge in cyber manufacturing systems is uniquely complex. The intricacies of cyber manufacturing systems make it impractical to directly apply recovery strategies proposed in the existing cyber-physical systems literature. Furthermore, given the significant cost of the assets within cyber manufacturing systems, the potential consequences of false data injection attacks leading the systems to unsafe states are catastrophic. To mitigate these risks and losses, an effective recovery strategy for a cyber manufacturing system must encompass several key objectives:

(1) It should successfully recover the impacted manufacturing process, ensuring that it returns to a normal operational state.
(2) The recovery strategy must ensure the continuous functioning of the manufacturing system, minimizing disruptions and production downtime.
(3) The strategy should avoid the need to shut down the impacted machines, as this can result in substantial production losses.
(4) The recovery plan should be designed to function without requiring additional hardware in the form of sensors, streamlining the implementation process and minimizing resource requirements.

In summary, the core problem addressed in this research revolves around developing a recovery strategy for cyber manufacturing systems that effectively addresses the unique challenges and requirements inherent to these systems, ultimately safeguarding their operational integrity and mitigating the potentially catastrophic consequences of false data injection attacks.

3. Design overview

In this section, the manufacturing system under consideration, threat modeling, and the design overview of the proposed recovery strategy are explained.

3.1. Manufacturing system

A manufacturing system is a complex and integrated framework that encompasses a range of processes, resources, and interactions, all working synchronously to produce the desired end product. In the development of such manufacturing systems, we have simulated an assembly line that comprises two robotic arms, a conveyor belt, and a drawing manufacturing process as depicted in Figure 3. These components operate in synchronization to manufacture products based on client specifications. The discrete events are simulated at the process level and operational/supervisory level.

In Figure 3 for the process level, there exists an independent controller designed to perform discrete process events. For example, the pick raw material discrete event at the operational level for robotic arm 1 has four process discrete events: idle, go to inventory, grab the box, and pick up the box. The operational/supervisory has eight discrete events: pick raw material, place raw material, moving conveyor to P2, hold conveyor for the manufacturing process, move conveyor to P3, pick finished product, move

conveyor to P3 and place finished product. The paper uses programmable logic controllers to sequence the operational/supervisory discrete events. To mimic the operations of future manufacturing systems, client orders are submitted through a web-based application developed in the Python programming language. Upon receiving these orders, they are stored in a local database and sequenced according to the first-in, first-out algorithm. This modeling approach enables to emulate the behavior of future manufacturing systems.

In this simulation, all physical systems, with the exception of the drawing process, are modeled using the discrete event simulation library SimPy in Python programming language, in conjunction with independent controllers and programmable logic controllers. The simulation is conducted over an 8-hour workday. Each step in the simulation represents a millisecond, signifying that every control decision made by the controller occurs at the millisecond level, ensuring precise control and monitoring of the manufacturing processes.

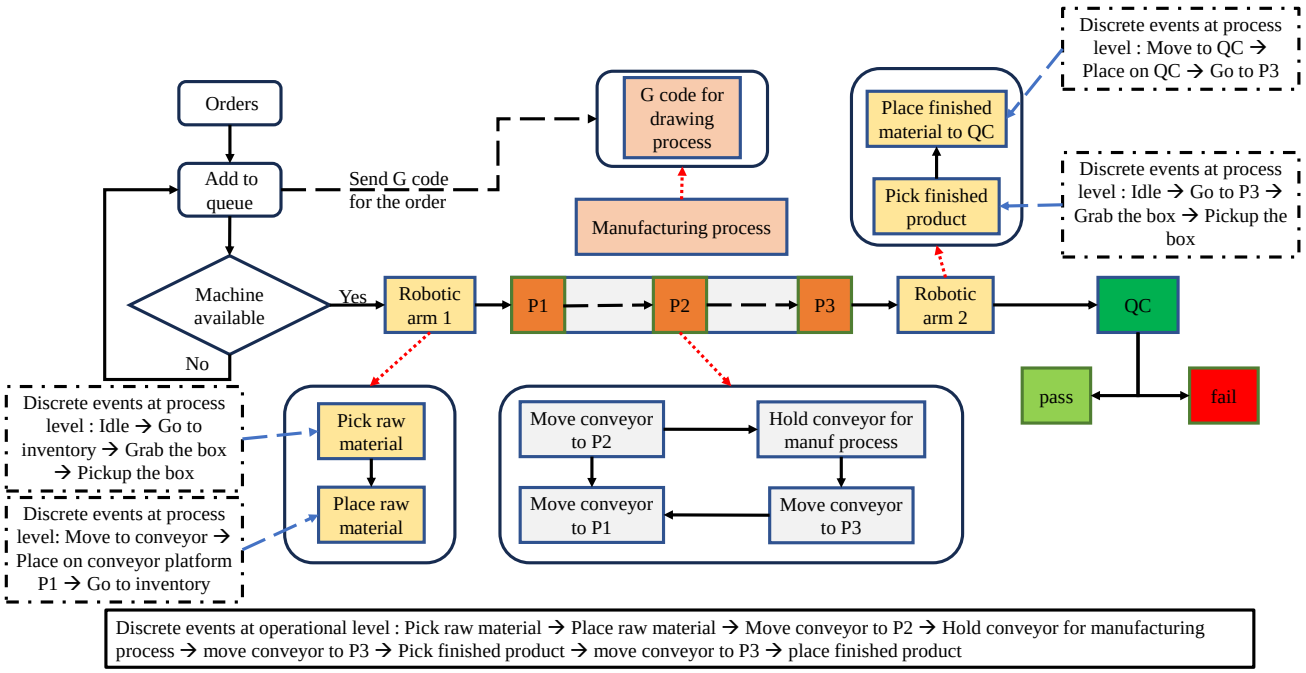

Figure 3. Manufacturing system used for experiment.

3.2. Threat modeling

This work considers malicious attackers capable of executing false data injection attacks on sensors. Specifically, this attacker can manipulate sensor data by introducing alteration before the data reaches the controller, thereby compromising the manufacturing system's integrity. The attack designed in this paper can randomly select one sensor from the simulator and initiate a false data injection attack by adding or subtracting the sensor measurement before processed by the controller. The attacker operates with two primary objectives in mind:

(1) **Threat 1: Driving the system towards physical limits**: One of the attacker's goals is to push the manufacturing system to its physical limits. The attacker achieves this goal by adding positive values to the sensor measurement. This can result in increased stress and potential wear and tear on the system's components, potentially causing long-term damage or reducing the system's operational lifespan.

(2) **Threat 2: Forcing system shutdown**: The attacker's second objective is to instigate a system shutdown. The attacker achieves this goal by adding negative values to the sensor measurement. The intent is to disrupt normal operations and potentially cause production downtime, leading to financial losses for the organization.

Both of these attacker objectives can have detrimental consequences for the manufacturing system, ranging from reduced equipment lifespan and operational inefficiencies to costly shutdowns and potential safety hazards.

3.3. Overview of the proposed recovery strategy

The proposed framework is illustrated in Figure 4. The operation of manufacturing systems consists of two modes: the default mode and the recovery mode. In the absence of an attack or when an attack remains undetected the manufacturing system operates on the default controller. However, upon detecting an attack, the controller is switched to a recovery mode. Within the recovery mode, there are two key controller components: the process level controller and a supervisor/operational-level controller.

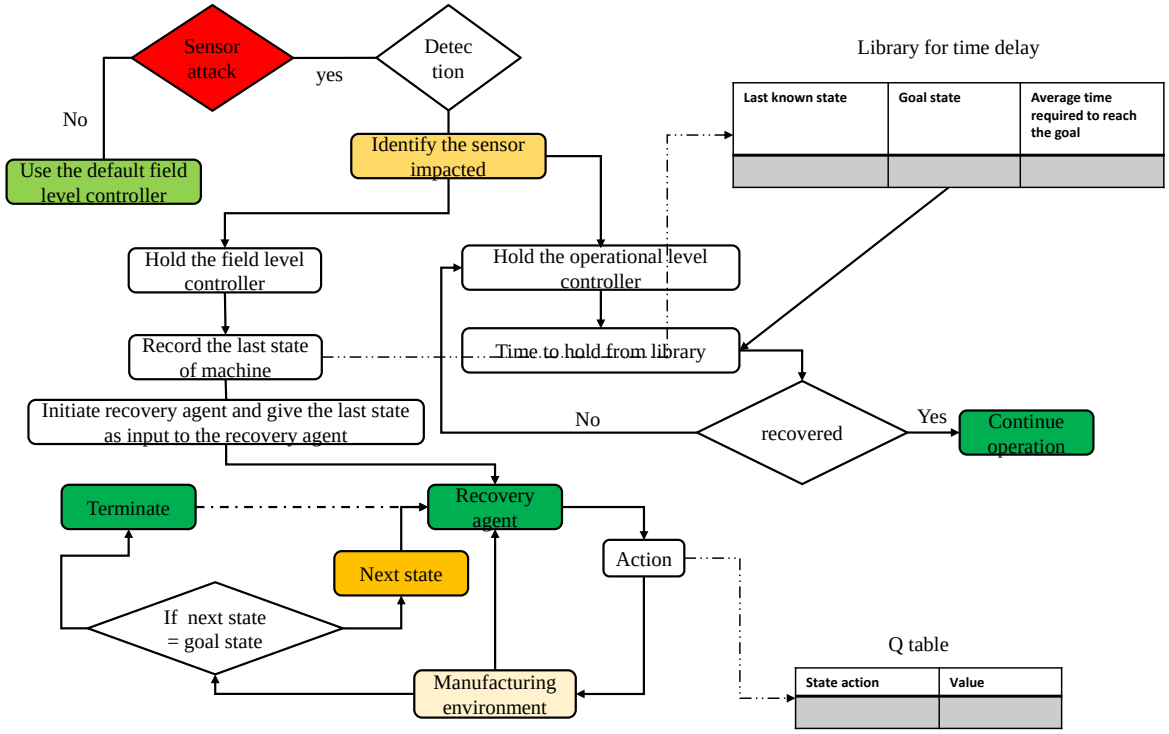

Figure 4. Recovery agent.

Upon detecting a false data injection attack on the sensor, the system generates an alert to point at the specific system and sensor under attack. The system then records the last state of the system and transmits this information to the recovery agent. This recovery agent is trained to navigate the system from its current position to a predefined goal position, aiming to restore the system's normal functioning. Meanwhile, the supervisory/operational controller operating on discrete events, delays the other process by the time proportional to the time required by the agent for system restoration. This approach allows the processes dependent on the impacted process to be in synchronization after the field-level recovery is achieved. Hence achieving an overall manufacturing systems recovery.

3.4. Challenges with the proposed work

The proposed work faces several challenges. Therefore, it is important to address them for the efficiency and reliability of the proposed recovery framework.

(1) **Incorrect last recorded state**: A significant challenge occurs when the last recorded state of the cyber manufacturing systems is incorrect due to the time delay between the attack occurring and detection [20]. During this delay the system may transition into undesired state, making the last recorded state unreliable for guiding recovery. To overcome this challenge the existing literature implemented a checkpoint protocol [14], recording the correct states and updating them with new correct states as the system moved forward in time. While this seems a reasonable approach when the final state or goal of the system is unknown. However, in the proposed work we can leverage the fact that the predefined operating conditions or goal states of the cyber manufacturing systems are known. By utilizing the system's known operating conditions, the agent can effectively drive recovery, minimizing the impact of sensor attacks and ensuring the system's resilience.

(2) **Avoiding redundant agents**: Ideally, each controller within the manufacturing system would require its own agent to handle recovery. However, this approach risks unnecessary redundancy, as multiple agents might be developed and trained to perform essentially the same tasks. To address this inefficiency, our solution employs a single, versatile agent trained to navigate various recovery scenarios across different processes. This unified agent is capable of guiding the impacted process back to its predefined goals, eliminating the need for multiple specialized agents. Meanwhile, controllers unaffected by the attack can continue to operate under the guidance of their default control algorithms, further simplifying the system and reducing computational overhead.

(3) **Recording time delay for operation/supervisory controller**: To identify the time required to delay the supervisory controller until the agent has recovered the system is achieved through library. During training, this architecture maintains a library that stores average time required by the agent to navigate from an unsafe state to safe state. When an attack is detected, the recovery agent is immediately deployed to control the compromised sensor. Simultaneously, the time delay from the library is applied to pause other sequential controllers until the agent completes its recovery tasks.

4. Reinforcement learning based recovery

4.1. Markov decision process

Developing a successful recovery strategy hinges on accurately estimating the future state based on the current state of the system, even when the current state is compromised due to a false data injection attack on the sensor. Despite the loss of information about the trajectory leading to the current state, the current state itself remains relevant for making predictions about the future state [21]. This characteristic aligns with the Markov property, indicating that the recovery environment can be treated as a Markov decision process (MDP). MDP is a mathematical framework widely used to model problems with discrete time horizons, signifying the presence of both a starting state and a termination state. The Markov property suggests that the future state of the system is conditionally independent of the past given the current state.

In the context of recovery from cyber attacks, the current state, though compromised, provides sufficient information for making predictions about subsequent states.

By formulating the recovery environment as an MDP, it becomes possible to apply MDP principles and methodologies to develop and optimize recovery strategies. This approach allows for a systematic and mathematically grounded exploration of decision-making processes, aiding in the design of effective recovery policies that take into account the dynamic nature of the system and the uncertainties introduced by the attacks. An MDP is defined by a five-state tuple (S, A, P, R, γ), where:

- S represents the set of states of the system s
- A is the set of control actions a
- P denotes the probability of transitioning from one state s to another state s+1
- r is the reward assigned when a state transition occurs
- γ is the discount factor, a value between 0 and 1. If γ is 0, only immediate rewards are considered, while if it's 1, rewards in future time steps are also taken into account.

$$r(s,a) = [R_{t+1} \mid S_t = s, A_t = a] \tag{1}$$

The primary objective of MDPs is to maximize expected rewards given by Equation (1). Two main approaches are commonly employed to solve this objective function: dynamic programming and reinforcement learning. Dynamic programming involves breaking the problem into simpler steps at different time points. Bellman's principle of optimality is a fundamental concept in dynamic programming [22]. It asserts that an optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision. In essence, it separates the current decision made at time step t from future decisions $t + 1$. Dynamic programming leverages recursion for expected rewards to find the optimal solution. However, a key challenge lies in determining the state transition probabilities, which can be computationally expensive, even when the system model is known [21]. This challenge often leads to the exploration of alternative approaches, such as reinforcement learning, which can be more practical and effective in some scenarios.

4.2. Reinforcement learning

Reinforcement learning has experienced a notable surge as an alternative control framework, offering a sample-based approach to address the limitations of traditional model-based approaches [21]. What sets reinforcement learning apart from other machine learning algorithms is its distinctive methodology—it constructs its own dataset through active exploration and exploitation of the environment. In this paradigm, an agent is trained to achieve a specific goal while receiving rewards for favorable actions and penalties for unfavorable ones. The agent's performance is solely assessed based on a scalar reward function, which quantifies the success or failure of its actions in the learning process. This unique characteristic of reinforcement learning makes it a powerful and adaptable tool, particularly suitable for scenarios where explicit models are challenging to define or compute. In the context of reinforcement learning the agent is represented as a 4-tuple (S, A, R, γ), where:

- S represents the set of states of the system s
- A is the set of control actions a

- R is the reward assigned when a state transition occurs
- γ is the discount factor, a value between 0 and 1. If γ is 0, only immediate rewards are considered, while if it's 1, rewards in future time steps are also taken into account.

The operation of reinforcement learning can be illustrated through a simplified example, as depicted in Figure 5. In this scenario, the agent, acting as the controller, aims to pick and place objects within a tray. At time t_0, the agent observes the state of the objects in the environment (S_t). Upon this observation, the agent selects an action from the set of control actions (A_t), which in this case involves adjusting the current or voltage to manipulate the robotic arm's motor. Following the execution of these actions, the agent receives a reward (R_{t+1}) determined during the modeling process, along with the next state of the system.

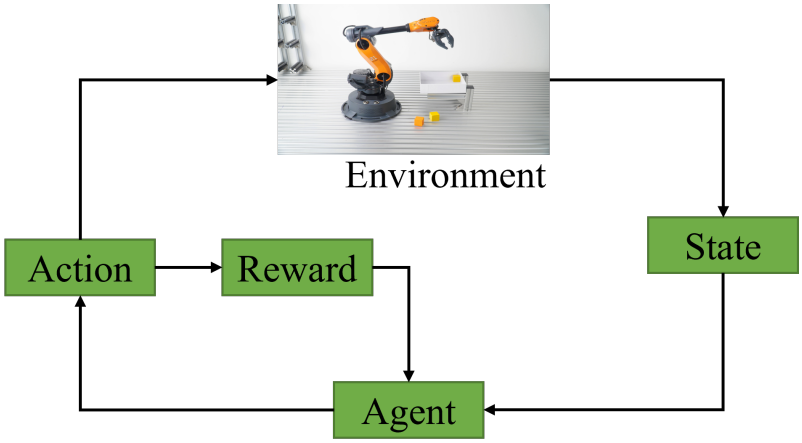

Figure 5. Toy example of reinforcement learning.

In each time step, the agent establishes a mapping from states to probabilities of selecting actions, known as the policy ($\pi_t(a|s)$). Reinforcement learning algorithms dictate how the agent adjusts its policy based on its experiences. Therefore, many reinforcement learning algorithms involve estimating the value function, as defined by Equation (2), or the state-action value function, as given by Equation (3). Estimating value functions enables the agent to understand the value of being in a particular state s following a policy π while estimating state-action value functions enables the agent to understand the value of taking a specific action a in state s following a policy π.

$$v_\pi(s) =_\pi \left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \mid S_t = s \right] \tag{2}$$

$$q_\pi(s,a) =_\pi \left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \mid S_t = s, A_t = a \right] \tag{3}$$

4.3. Reinforcement learning based recovery

The reinforcement learning framework serves as a powerful abstraction for addressing the challenge of goal-directed learning through interaction [21]. In the context of this paper, the objective of the recovery agent is to navigate the cyber-attacked manufacturing system back to its predefined conditions, making the reinforcement learning framework well-suited for this problem.

4.3.1 Characteristics of the agent

The crucial elements of observation, rewards, and actions are defined as follows:

Environment: The manufacturing system simulation consists of two robotic arm, one conveyor and a manufacturing process. This constitutes the agent's environment. Additionally, the sensors providing information on positions, current, and the status of discrete events within the plant are integral components of the environment. To simulate real-world imperfections, Gaussian noise in the range of $[-0.2, 0.2]$ is added to the sensor data, representing potential measurement inaccuracies and ensuring the system is robust against minor perturbations. Table 1 shows the detailed environment the agent observes. In the table, the discrete events are represented as binary information.

Table 1. Agent's environment.

Machines	**Observations**
Robotic arm 1	$\{P_1, \dots P_6\}$ in degree
	$\{I_1, \dots I_6\}$ in mA
Robotic arm 2	$\{P_1, \dots P_5\}$ in degree
	$\{I_1, \dots I_5\}$ in mA
Conveyor belt	$\{P_1, P_2, P_3\}$ in mm
	$\{I_1\}$ in mA
Manufacturing process	Time to process in s
Discrete events	R1 pickup object: {1,0}
	R1 drop object: {1,0}
	Conv moves to P2 position: {1,0}
	Manufacturing process: {1,0}
	Conv moves to P3 position: {1,0}
	R2 pickup finished product: {1,0}
	Conv moves to P1 position: {1,0}
	R2 drop finished product for QC: {1,0}

State/Observation Space: Considering the complexity of the environment, it is essential to provide the agent with a subset of information as its state/observation space. In this paper, the agent observes the information only for the sensor detected by the detection system. Additionally, the agent receives information about the discrete event to which the detection belongs. For example, if the conv moves to P2 position is currently in motion then after detection the agent will receive: [(Discrete events: [0,0,1,0,0,0,0,0]),(current state of the conveyor: P_1'). In Table 1 P_1 is the true state, however since after the attack the state is not true and hence represented as P_1', (Goal state: P_2)].

Action Space: The action space is straightforward, encompassing discrete control steps such as increasing the current, decreasing the current, or maintaining it at a constant level. In the environment, this is represented as 0,1 and 2 numbers. However inside the controller simulator, number 0 corresponds to decreasing the current, 1 corresponds to constant, and 2 corresponds to increasing the current.

Reward: Given the goal-directed nature of the recovery agent, any state other than the goal state is deemed incorrect. The reward value given in Equation (4) is based on the difference (r) between the goal state (T_{goal}) and the actual state of the system (T_{act}). If this difference falls within the range $[-2, 2]$—representing an acceptable tolerance for noise and small deviations in manufacturing systems—the agent receives a positive reward of 5, signaling successful recovery given by Equation (5). However, if r falls outside this range, the agent incurs a negative reward, penalizing deviations from the

predefined goal state.

The choice of assigning a reward value of 5 in the proposed work is directly linked to the physical characteristics of the system, specifically the motor's operation, where increments occur in 5-degree steps. This alignment ensures that the reward structure is both meaningful and consistent with the practical behavior of the manufacturing system. By using the motor's incremental step size as the basis for the reward, the reinforcement learning agent's feedback mechanism is closely tied to the resolution of the recovery task it is designed to address.

$$r = T_{goal} - T_{act} \tag{4}$$

where:

$$\begin{cases} r = 5 & -2 \leq r \leq 2 \\ r = -\mid r \mid & else \end{cases} \tag{5}$$

Algorithm: The proposed work implements a Q-learning algorithm [23], selected over advanced algorithms like proximal policy optimization (PPO) [24] and deep Q-network (DQN) [25] due to its simplicity and computational efficiency. Q-learning's lightweight design makes it ideal for cyber manufacturing systems, which often lack high-performance hardware such as GPUs (Graphics Processing Unit) required for training complex models. Additionally, its tabular approach provides a clear and interpretable state-action mapping, which is advantageous in ensuring transparency and reliability in recovery strategies. Unlike PPO and DQN, Q-learning does not require extensive hyperparameter tuning, reducing training complexity and time. Moreover, its hardware independence simplifies integration with existing manufacturing systems without necessitating additional computational resources, aligning with the goal of avoiding extra hardware requirements in the recovery plan. It is a model-free algorithm and learns the value of an action in a particular state by using equation 6. In this equation Q(s, a) represents the Q-value for state-action pair (s, a), α is the learning rate, r is the immediate reward, γ is the discount factor, s' is the next state, and a' is the action in the next state. The Q table is updated as the agent explores and exploits the environment.

$$Q(s,a) = (1-\alpha) * Q(s,a) + \alpha * [r + \gamma * max(Q(s',a'))] \tag{6}$$

4.3.2 Recovery agent

The development of the recovery agent, depicted in Figure 6 is an enlarged view of the recovery agent from Figure 4. It plays a crucial role in restoring the cyber-attacked manufacturing system to its desired state. At the heart of this process is the Q-learning algorithm, an integral part of reinforcement learning that enables the agent to learn a policy, mapping states to actions, through iterative exploration and exploitation. The agent initiates its recovery process from the last recorded state of the system, serving as its starting point. The primary objective of the recovery agent is to navigate the system back to its predefined operating conditions, overcoming the disruptions caused by cyber-attacks.

The Q-learning algorithm is employed iteratively until the agent successfully reaches its goal. Concurrently, a Q table is constructed to store the learned values that guide the agent's decision-making process. Simultaneously, a library is developed to track the average time required by the agent to reach its

goal during the learning process. This library is utilized by the supervisory/operational level recovery. During the training phase, the starting state of the system is randomly selected from the predefined physical limits associated with each actuator. This randomness in the selection process ensures that the agent encounters a diverse set of starting conditions, mirroring the potential scenarios it might face in real-world applications. This approach helps the agent generalize its learning, making it robust across various initial states.

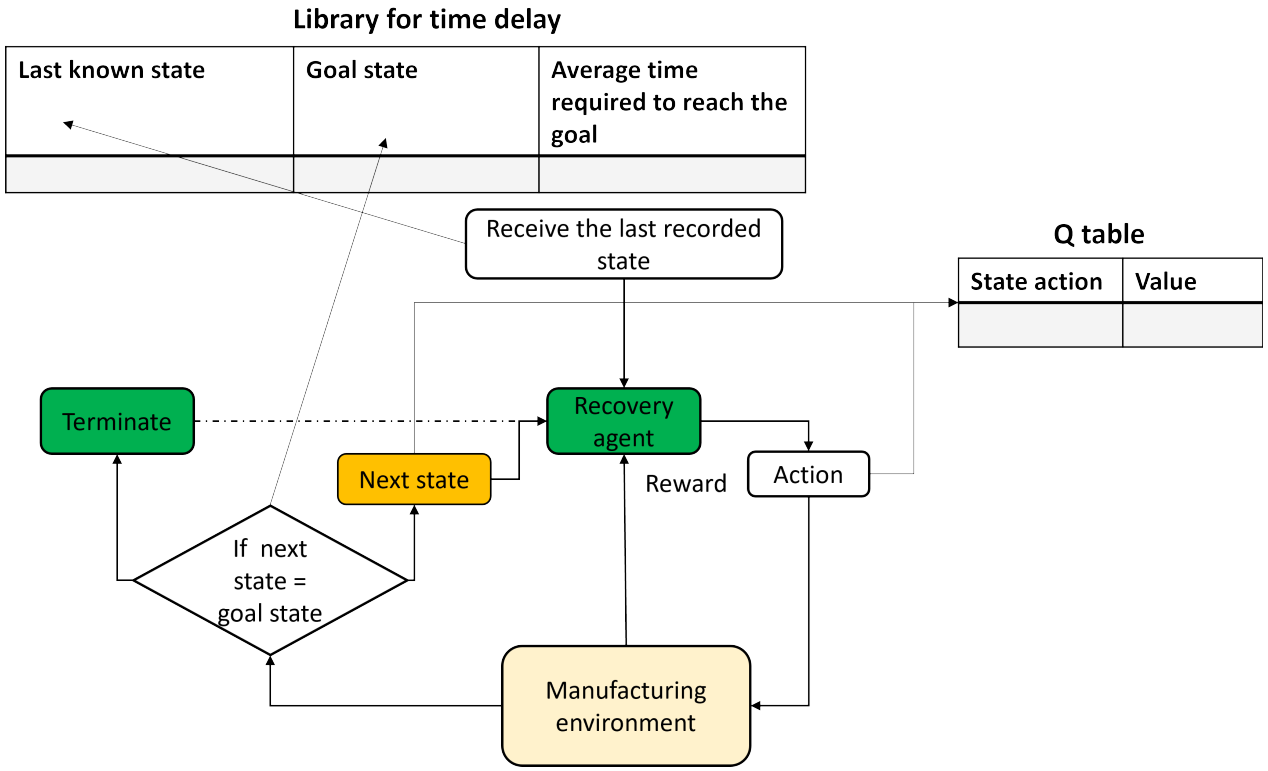

Figure 6. Reinforcement learning-based recovery agent (Enlarged view of the Figure 4).

5. Results and discussion

To assess the performance of the recovery strategies outlined in this paper, comparisons are made against two reference benchmarks: (i) restarting the process and (ii) a manually tuned PID (Proportional-Integral-Derivative) controller. Restarting the system is a straight forward approach to eliminate the effects of attacks. PID controller, on the other hand, offers a manual and more reliable way to recover systems by guiding the system to a predefined safe checkpoint before resuming normal operation. This comparative analysis will provide valuable insights into the effectiveness of the proposed recovery strategies, enabling manufacturers to make informed decisions about their adoption and implementation.

5.1. Recovery from threat 1

Threat 1 involves false data injection attacks on sensors, pushing the manufacturing system towards its physical limits. In Figure 7, the recovery strategies of the recovery agent and PID controller are demonstrated as they restore the system states to normal conditions. The PID controller's approach comprises a two-step recovery process. Initially, it brings the system back to a checkpoint state,

representing the last known safe state, before resuming normal operation. This two-step recovery strategy ensures stability but often at the expense of speed and precision. The PID controller's performance depends heavily on accurate manual tuning and predefined parameters, such as the location of the safe checkpoints. In dynamic manufacturing systems where operating conditions or attack scenarios vary, this manual intervention can be time-intensive and prone to errors. Additionally, the reliance on checkpoints makes the PID controller less adaptable to new and unforeseen scenarios, limiting its robustness in environments where flexibility is critical.

In contrast,the RL-based recovery agent offers a significant improvement by taking a direct path from the impacted state to the predefined goal state, bypassing intermediate checkpoints. This streamlined approach along with the time delay library minimizes recovery time, as evident in the results in Figure 8 and Figure 9, where the RL agent outperforms the PID controller in restoring system states. The time delay library ensures that the discrete event is paused until the agent has reached the safe state. The RL agent's ability to learn and generalize from diverse training scenarios equips it to handle various attack-induced disruptions, making it more adaptable and robust than the PID controller. Moreover, the RL agent eliminates the need for manual tuning and pre-specified safe checkpoints, significantly reducing the complexity of implementation. Its reliance on a simple computational framework further enhances its practicality, as it does not require advanced hardware or GPU resources, ensuring compatibility with existing manufacturing systems.

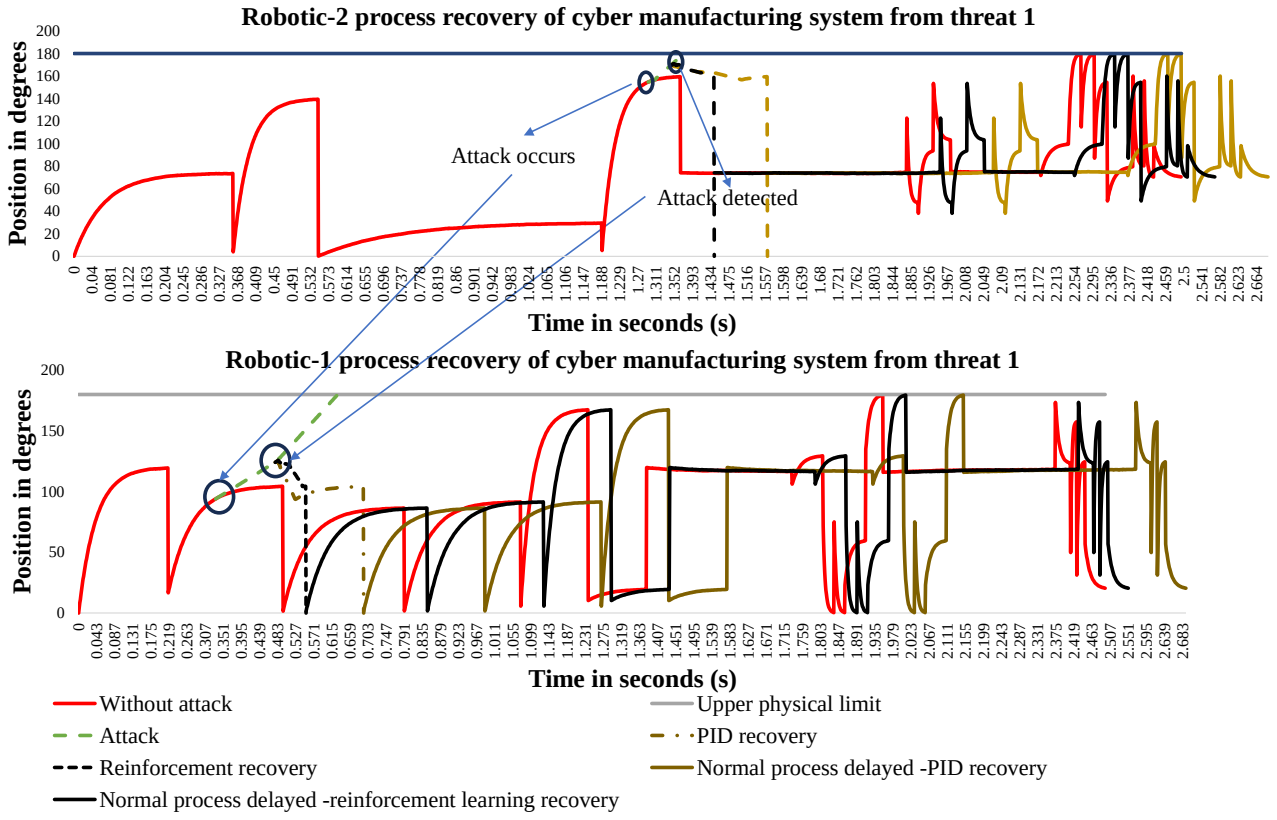

Figure 7. Process level recovery of cyber manufacturing system from false data injection attacks on sensors (Threat 1).

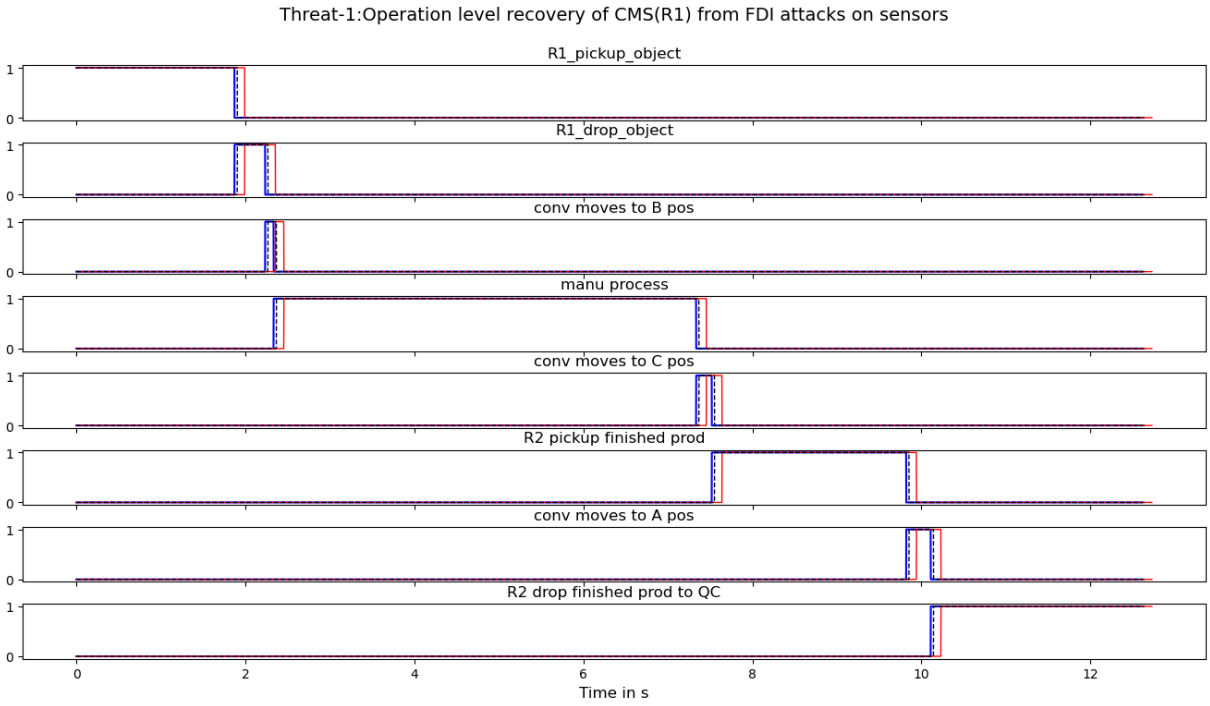

Figure 8. Operation level recovery of cyber manufacturing system (R1) from false data injection attacks on sensors (Threat 1). The blue color represents the normal functioning of manufacturing systems. The black color represents recovery with reinforcement learning and the red color represents recovery with PID.

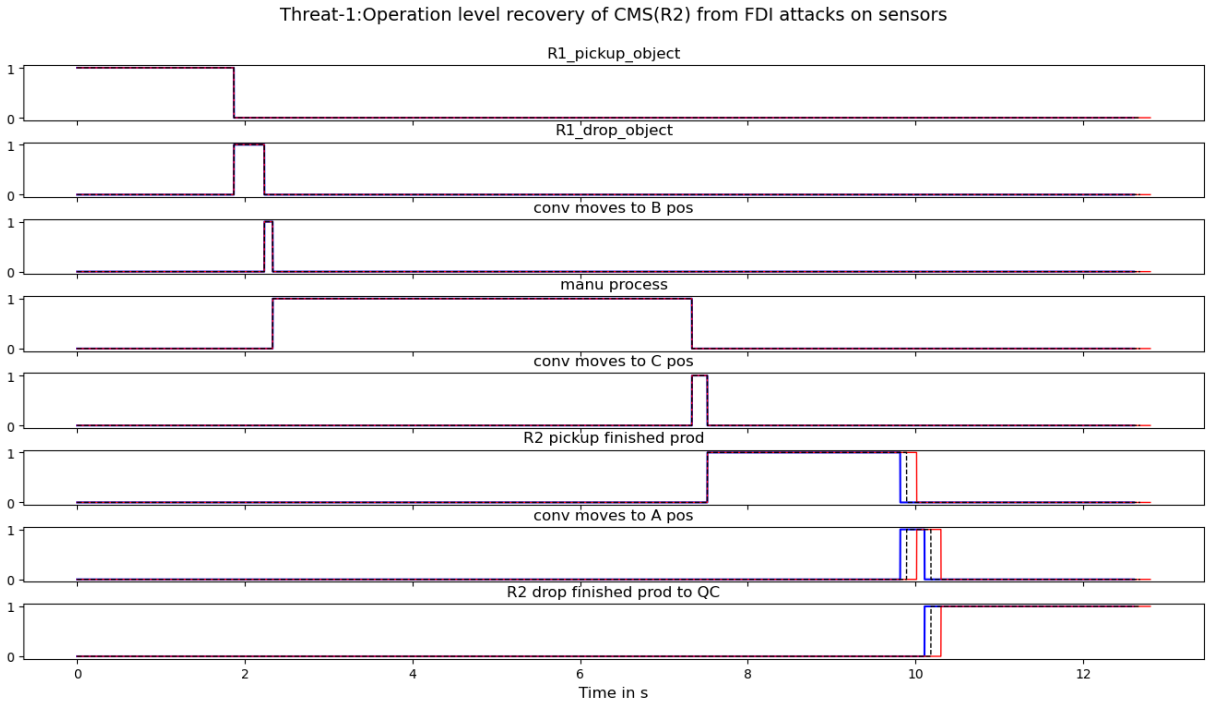

Figure 9. Operation level recovery of cyber manufacturing system (R2) from false data injection attacks on sensors (Threat 1). The blue color represents the normal functioning of manufacturing systems. The black color represents recovery with reinforcement learning and the red color represents recovery with PID.

5.2. Recovery from threat 2

Threat 2, as depicted in Figure 10, represents a scenario where false data injection attacks aim to push the system toward zero, mimicking a shutdown state that could result in severe disruptions to manufacturing operations. A closer analysis of the graph highlights several key advantages of the RL agent. First, the RL agent adapts dynamically to the state of the system, enabling it to identify and follow the most efficient path back to the desired operational state. This contrasts with the PID controller, which takes a longer recovery path as it first stabilizes the system at an intermediate checkpoint before progressing to full recovery. In practical terms, this difference translates to reduced recovery times for the RL agent, which minimizes production downtime and ensures continuity of operations—a critical consideration in manufacturing environments where even brief interruptions can lead to significant financial losses.

Additionally, the RL agent leverages the time-delay library to effectively manage the timing of discrete events during recovery as depicted in Figure 11. This feature ensures that the system's discrete controllers remain synchronized with the recovery process, preventing further disruptions or cascading errors that could exacerbate the attack's impact. The PID controller, lacking such a mechanism, is more prone to delays caused by misaligned event timing, particularly in complex scenarios like threat 2, where precise coordination is crucial.

The RL agent's efficiency is particularly advantageous in manufacturing systems where downtime is costly. For instance, in industries with tightly scheduled production lines or high operational throughput, even a minor delay in recovery can translate into significant losses in revenue or missed delivery deadlines. By offering a faster, automated, and dynamic recovery mechanism, the RL agent reduces these risks while eliminating the need for manual intervention, hardware upgrades, or system restarts.

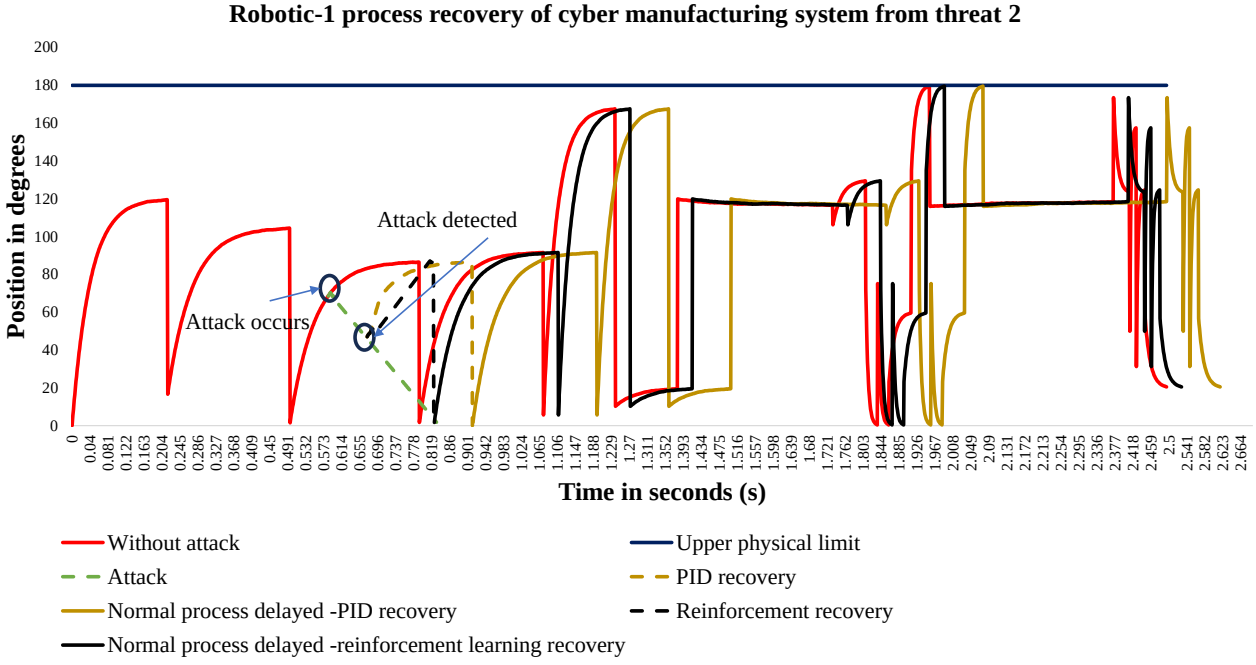

Figure 10. Process level recovery of cyber manufacturing system from false data injection attacks on sensors (Threat 2).

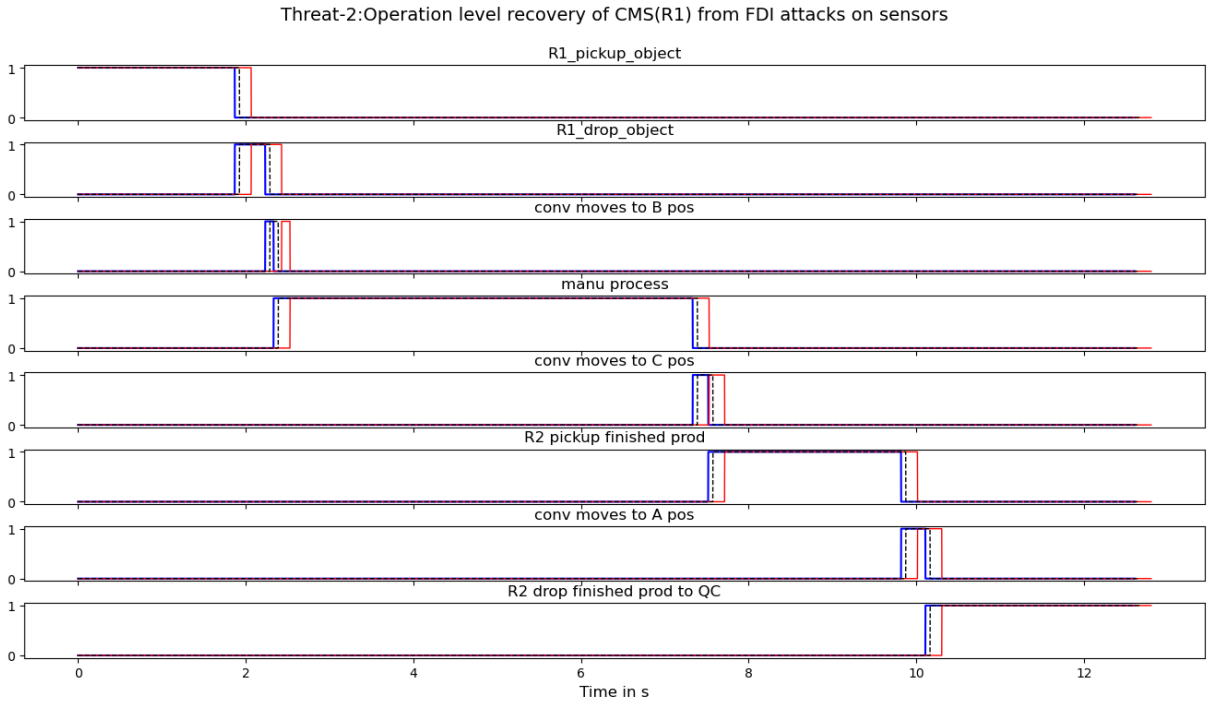

Figure 11. Operation level recovery of cyber manufacturing system (R1) from false data injection attacks on sensors (Threat 2). The blue color represents the normal functioning of manufacturing systems. The black color represents recovery with reinforcement learning and the red color represents recovery with PID.

5.3. Metrics to validate the agent

However, validating the performance of the recovery strategies presented in this paper requires the establishment of key metrics. Three primary metrics have been selected for this purpose, drawing inspiration from traditional key performance indicators commonly used by manufacturing plants. These metrics are essential for evaluating the effectiveness of the recovery strategies and making informed comparisons. The three chosen metrics are:

Downtime: Downtime represents the total time during which the manufacturing process is non-operational, often due to disruptions or issues, and is given by Equation (7). Minimizing downtime is a critical objective in manufacturing, as it directly impacts production efficiency and profitability. The recovery strategies introduced in this paper will be assessed based on their ability to reduce downtime.

$$Downtime = \frac{\text{Downtime}}{\text{Downtime} + \text{Uptime}} \tag{7}$$

Efficiency: Efficiency reflects how well resources are utilized in the manufacturing process to produce the desired output and is given by Equation (8). Maximizing efficiency is a fundamental objective for manufacturing operations, as it directly impacts production costs and resource allocation. The proposed recovery strategies will be analyzed for their efficiency improvements.

$$Efficiency = \frac{\text{Actual cycle time}}{\text{Expected cycle time}} \tag{8}$$

Throughput: Throughput represents the rate at which products are processed or manufactured within a given time frame and is given by Equation (9). It's a fundamental indicator of production capacity and efficiency. Maximizing throughput is a key goal for manufacturers, as it directly impacts the rate of production and, ultimately, revenue generation. The recovery strategies will be analyzed in terms of their impact on throughput improvement.

$$Throughput = \frac{\text{Units produced}}{\text{Time}} \tag{9}$$

The Figure 12 illustrates throughput and efficiency during an 8-hour shift under different recovery scenarios. The proposed recovery method, powered by reinforcement learning (RL), demonstrates a throughput nearly equal to the "normal no attack" condition, signifying its ability to restore production to pre-attack levels efficiently. In contrast, the PID controller yields significantly lower throughput. This disparity stems from the PID method's reliance on manual tuning and a multi-step recovery process, which delays a full return to normal operations. Restarting the system fares better than PID, as it neutralizes the attack entirely, but its throughput remains lower than the RL approach due to the inherent delay from the system's reboot and startup sequence. This comparison underscores the RL agent's resilience, as it mitigates the attack's impact while enabling continued production.

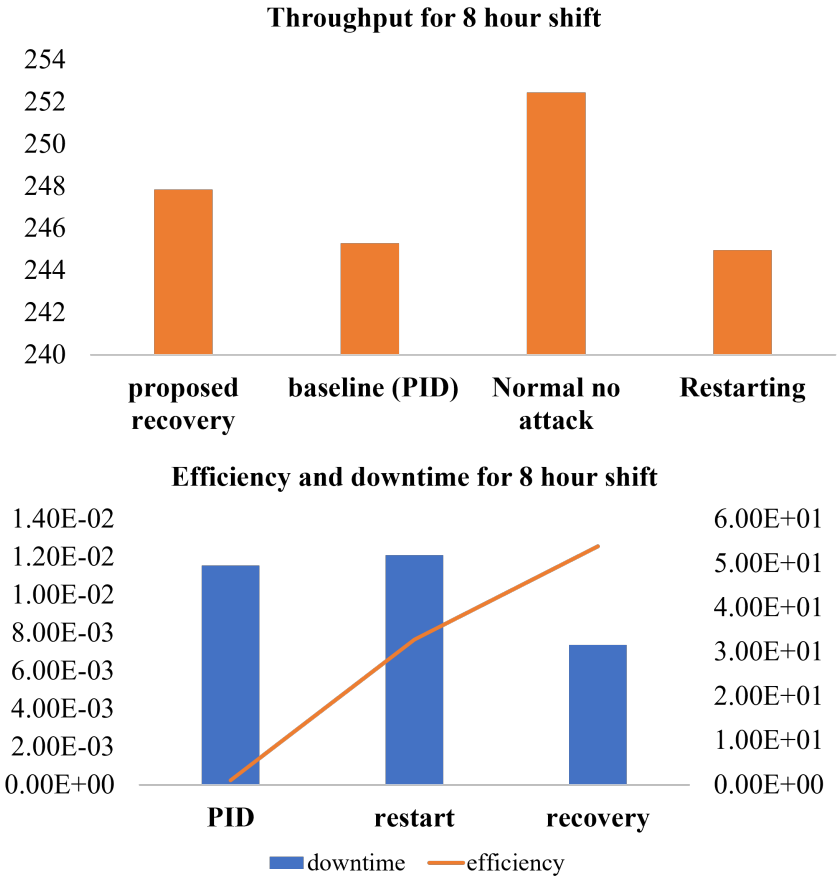

Figure 12. Metrics to evaluate the performance of the recovery agent.

In the lower section of the figure, efficiency and downtime are depicted on a dual-axis chart to show their interplay under different recovery strategies. The downtime metric (shown as blue bars) reveals

that the PID controller results in the most prolonged disruptions due to its slower, iterative recovery process. Consequently, its efficiency (depicted as the orange line) is the lowest, reflecting poor resource utilization and substantial production loss. System restarting exhibits moderate downtime, as it eliminates the attack's effects but requires significant time for the reboot sequence, which limits its overall efficiency. The proposed RL-based recovery, however, achieves the shortest downtime, as it rapidly restores the system to normal conditions without the need for a full reset or manual intervention. This minimized downtime directly contributes to the highest efficiency of the RL approach, indicating optimal use of manufacturing resources and minimal wastage during recovery.

The key insight from this analysis is that the recovery architecture addresses all four key questions, demonstrating its robustness and efficiency. First, it successfully restores the impacted manufacturing process to a normal operational state, as evidenced by the recovery illustrated in Figure 7 and 10. Second, the strategy ensures the continuous functioning of the manufacturing system by minimizing disruptions and downtime, as highlighted in Figure 8, 9, and 11. Third, the approach avoids shutting down impacted machines, mitigating substantial production losses compared to shutdown methods, as shown in Figure 12. Lastly, the recovery plan operates without requiring additional hardware, leveraging the lightweight Q-learning algorithm to utilize existing sensors and resources effectively. These findings underscore the superiority of the RL-based recovery strategy in enabling efficient, adaptive, and resource-conscious recovery for manufacturing systems.

6. Discussion

6.1. Scalability

The proposed recovery solution is highly scalable, making it suitable for large-scale, sensor-dense manufacturing systems. By employing a Q-learning-based recovery algorithm that operates on a per-sensor basis, the framework ensures localized decision-making, minimizing computational overhead even in environments with thousands of sensors. The modular design allows the system to monitor all sensors in parallel but activate recovery protocols only for the affected sensor, optimizing resource utilization. While the current approach addresses single-sensor failures, it can be extended to handle multiple simultaneous failures using multi-agent systems, where each agent independently manages a subset of sensors. Furthermore, the lightweight nature of Q-learning ensures minimal resource requirements, maintaining feasibility in resource-constrained settings.

6.2. Reliance on detection system

The success of the proposed work is heavily dependent on the accuracy of the detection system. The recovery agent only recovers the system after detection system identify the attacked sensor. Choosing an appropriate detection system introduces several challenges, as there is often a tradeoff between achieving higher true positives and minimizing false negatives. The effectiveness of the overall recovery strategy hinges on the reliability and precision of the detection system in identifying and alerting the presence of false data injection attacks on sensors.

6.3. Stability of the cyber manufacturing system

The challenge of defining stability in control systems, especially in the context of reinforcement learning (RL), is indeed a complex task. Traditionally, Lyapunov functions have been employed to analyze the stability of dynamical systems [26]. However, the lack of a clear mathematical model and understanding of the RL algorithms poses a unique set of challenges in establishing stability for RL-based control systems. One potential avenue for assessing stability in RL applications is to examine the convergence of rewards. A stable RL algorithm should exhibit consistent and convergent reward values over time. Tracking the trend of rewards during training and ensuring that they stabilize within an acceptable range can provide insights into the stability of the learned policy.

Moreover, the trade-off between exploitation and exploration in RL is a crucial factor. A stable RL algorithm should strike a balance between exploiting the learned policy and exploring new actions to adapt to changes or uncertainties. Monitoring this trade-off and ensuring that the agent converges to a robust policy without drastic fluctuations is indicative of stability. Additionally, leveraging insights from control theory and adapting them to the unique characteristics of RL algorithms could contribute to addressing the stability concerns in these complex cyber-physical systems.

6.4. Multiple sensor attacked at the same time

As of now, this work focuses solely on addressing a single attack on the system. However, in scenarios where multiple attacks occur simultaneously, the proposed strategy would need to be adapted to incorporate a multi-agent simulation. This adjustment becomes crucial to ensure the effectiveness of the recovery strategy in the face of multiple concurrent attacks on the cyber manufacturing system.

6.5. Complexity of the simulator

While the proposed simulation serves as a promising proof of concept, it's essential to acknowledge that real-world manufacturing scenarios involve not just a singular process but multiple processes running in parallel. Expanding the scope of this work to accommodate and address the challenges posed by multiple parallel processes would enhance its applicability and relevance in complex manufacturing environments.

6.6. Sample inefficiency and real-world transfer challenges

Reinforcement learning indeed faces the challenge of sample inefficiency, particularly in scenarios with limited and repetitive real-world data. In the manufacturing domain, where patterns in data can be highly similar, training an efficient agent becomes a challenging task due to the scarcity of diverse samples. To overcome this hurdle, it is crucial to explore strategies for improving sample efficiency. Techniques such as data augmentation, ensemble learning, or leveraging domain knowledge to generate synthetic data could be explored.

Furthermore, transferring the learned policies from simulation to the real-world manufacturing testbed is a significant challenge. Discrepancies between the simulated environment and the actual system can lead to a lack of generalization. Addressing this issue may involve refining the simulation model to better match the real-world dynamics or adopting techniques like domain adaptation to bridge the gap

between simulation and reality. Another critical consideration is synchronizing the decisions made by the reinforcement learning agent with the hardware's clock timing in the manufacturing system. Ensuring that the agent's decisions align seamlessly with the physical processes is essential for the successful implementation of the proposed recovery strategy. Fine-tuning the agent's temporal aspects and addressing any timing mismatches are crucial steps in achieving effective real-world deployment.

6.7. Computational cost

From a computational perspective, the initial training phase can be resource-intensive due to the iterative nature of RL algorithms and the need for large numbers of episodes to converge on an optimal policy. However, once trained, the agent operates efficiently, requiring minimal computational overhead during real-time deployment, making it suitable for resource-constrained environments. The costs associated with implementation include setting up a reliable detection system, maintaining the RL agent's training environment, and periodically updating the model to adapt to changes in system dynamics or attack patterns. While these costs may be non-trivial, they are offset by the system's ability to minimize downtime and enhance operational resilience, leading to long-term benefits in productivity and security. Balancing these trade-offs is essential for the successful adoption of the RL-based recovery strategy in manufacturing systems.

7. Conclusion

In the realm of resilient manufacturing systems, this paper brings significant contributions by addressing the critical aspect of recovery. While existing literature predominantly emphasizes prevention and detection strategies, the proposed work focuses on recovery. The key contributions of this paper can be summarized as follows. This paper pioneers the development of a robust recovery strategy for cyber manufacturing systems. The approach leverages reinforcement learning to guide the system back to its normal operating conditions after being subjected to false data injection attacks on sensors. By utilizing the principles of reinforcement learning, the recovery process becomes adaptive, learning from the system's environment and efficiently navigating toward normalcy. Using a manufacturing system simulator this work demonstrates that our proposed work can successfully recover the system and ensure that it continues its functioning by minimizing the downtime, and outperforms the manually tuned PID controller. By contributing pioneering methodologies for recovery and continuous operation, this paper significantly advances the field of resilient manufacturing systems. It addresses a critical gap in existing research and provides practical insights into enhancing the overall robustness and adaptability of cyber-physical manufacturing environments.

Acknowledgments

This project is not funded by any external source.

Conflicts of Interests

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Authors contribution

Romesh Prasad: Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing—original draft, review and editing, Visualization. Young Moon: Conceptualization, Resources, Writing—review and editing, Supervision, Project administration.

References

[1] Yampolskiy M, King WE, Gatlin J, Belikovetsky S, Brown A, *et al.* Security of additive manufacturing: Attack taxonomy and survey. *Addit. Manuf.* 2018, 21:431–457.

[2] Worley M, Caridi C, Alvarez M, Bakken K, Bedard Y, *et al.* X-Force Threat Intelligence Index. Technical Report, IBM, 2023. Available: https://secure-iss.com/wp-content/uploads/2023/02/IBM-Security-X-Force-Threat-Intelligence-Index-2023.pdf (accessed on 6 October 2024).

[3] Salazar LAC, Alvarado OAR. The future of industrial automation and IEC 614993 standard. In *2014 III International Congress of Engineering Mechatronics and Automation (CIIMA)*, Cartagena, Colombia, October 22–24, 2014, pp. 1–5.

[4] Cruz Salazar LA, Ryashentseva D, Lüder A, Vogel-Heuser B. Cyber-physical production systems architecture based on multi-agent's design pattern—comparison of selected approaches mapping four agent patterns. *Int. J. Adv. Manuf. Technol.* 2019, 105(9):4005–4034.

[5] Brandman J, Sturm L, White J, Williams C. A physical hash for preventing and detecting cyber-physical attacks in additive manufacturing systems. *J. Manuf. Syst.* 2020, 56:202–212.

[6] Krundyshev V, Kalinin M. Prevention of cyber attacks in smart manufacturing applying modern neural network methods. *IOP Conf. Ser.: Mater. Sci. Eng.* 2020, 940(1):012011.

[7] Yu Z, Zhou L, Ma Z, El-Meligy MA. Trustworthiness modeling and analysis of cyber-physical manufacturing systems. *IEEE Access* 2017, 5:26076–26085.

[8] Masuda AS. Cyber-Physical Attack Detection and Localization in Additive Manufacturing Systems. Ph.D. Thesis, University of California, Irvine, 2023.

[9] Rahman MH, Shafae M. Physics-based detection of cyber-attacks in manufacturing systems: A machining case study. *J. Manuf. Syst.* 2022, 64:676–683.

[10] Wu M, Moon YB. Intrusion detection system for cyber-manufacturing system. *J. Manuf. Sci. Eng.* 2019, 141(3):031007.

[11] Wu M, Moon YB. Alert correlation for detecting cyber-manufacturing attacks and intrusions. *J. Comput. Inf. Sci. Eng.* 2020, 20(1):011004.

[12] Yu SY, Malawade AV, Chhetri SR, Al Faruque MA. Sabotage attack detection for additive manufacturing systems. *IEEE Access* 2020, 8:27218–27231.

[13] National Institute of Standards and Technology. The NIST Cybersecurity Framework (CSF) 2.0. Technical Report, 2023. Available: https://nvlpubs.nist.gov/nistpubs/CSWP/NIST.CSWP.29.pdf (accessed on 6 October 2024).

[14] Kong F, Xu M, Weimer J, Sokolsky O, Lee I. Cyber-physical system checkpointing and recovery. In *2018 ACM/IEEE 9th International Conference on Cyber-Physical Systems (ICCPS)*, Porto, Portugal, April 11–13, 2018, pp. 22–31.

[15] Huang J, Chang Q, Arinez J. Deep reinforcement learning based preventive maintenance policy for

serial production lines. *Expert Syst. Appl.* 2020, 160:113701.

[16] Langner R. Stuxnet: Dissecting a cyberwarfare weapon. *IEEE Security & Privacy* 2011, 9(3):49–51.

[17] Zhang L, Lu P, Kong F, Chen X, Sokolsky O, *et al.* Real-time attack-recovery for cyber-physical systems using linear-quadratic regulator. *ACM Trans. Embedded Comput. Syst.* 2021, 20(5s):1–24.

[18] Zhang L, Sridhar K, Liu M, Lu P, Chen X, *et al.* Real-Time Data-Predictive Attack-Recovery for Complex Cyber-Physical Systems. In *2023 IEEE 29th Real-Time and Embedded Technology and Applications Symposium (RTAS)*, San Antonio, USA, May 9–12, 2023, pp. 209–222.

[19] Akowuah F, Prasad R, Espinoza CO, Kong F. Recovery-by-learning: Restoring autonomous cyber-physical systems from sensor attacks. In *2021 IEEE 27th International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA)*, Houston, USA, August 18–20, 2021, pp. 61–66.

[20] Zhang L, Chen X, Kong F, Cardenas AA. Real-time attack-recovery for cyber-physical systems using linear approximations. In *2020 IEEE Real-Time Systems Symposium (RTSS)*, Houston, USA, December 1–4, 2020, pp. 205–217.

[21] Sutton RS, Barto AG. *Reinforcement learning: An introduction*. Cambridge: MIT press, 2018.

[22] Kirk DE. *Optimal Control Theory: An Introduction*, Hoboken: Prentice-Hall, 1970.

[23] Dietterich TG. Hierarchical reinforcement learning with the MAXQ value function decomposition. *J. Artif. Intell. Res.* 2000, 13:227–303.

[24] Schulman J, Wolski F, Dhariwal P, Radford A, Klimov O. Proximal policy optimization algorithms. *arXiv* 2017, arXiv:1707.06347.

[25] Mnih V. Playing atari with deep reinforcement learning. *arXiv* 2013, arXiv:1312.5602.

[26] Lyapunov AM. The general problem of the stability of motion. *Int. J. Control* 1992, 55(3):531–534.