SSRG International Journal of Electrical and Electronics Engineering
ISSN: 2348-8379/ https://doi.org/10.14445/23488379/IJEEE-V13I9P104

Volume 13 Issue 9, 43-54, September 2026

Original Article

Agentic AI Controller for Autonomous Industrial Network Recovery

A. Kumaraswamy[1], G. Shailaja[2], Tanmaya Bhoi[3], Prashant Sharma[4], T.B Sivakumar[5], Umang Gupta[6]

[1]*Mechanical Engineering, Sri Venkateswara College of Engineering, Sriperumbudur Tk, Chennai, Tamil Nadu, India.*
[2]*Department of IT, CVR College of Engineering, Hyderabad, Telangana, India.*
[3]*Computer Science, Rajendra University, Prajna Vihar, Balangir, Odisha, India,*
[4]*Computer science & Engineering, University of Rajasthan, Jaipur, India.*
[5]*Department of CSE, School of Computing, Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, Chennai, Tamil Nadu, India.*
[6]*Business Analytics, Graduate school of business, Tula's UNIVERSITY, Dhoolkot, P.O. Selaqui, Chakrata Road, Dehradun, Uttarakhand, India.*

[1]*Corresponding Author : kumaraswamya2506@gmail.com*

Received: 27 July 2026 Revised: 08 September 2026 Accepted: 17 September 2026 Published: 26 September 2026

Abstract - *Industrial networks must ensure that there is a path to re-establish communications and recover from incident failures, congestion, communication failures, and controller outages without violating process deadlines or security segmentation. This paper presents AAC-AINR, a new agentic AI controller that enables topology-aware diagnosis, retrieval-grounded planning, digital-twin validation, transactional Software-Defined Networking (SDN) execution and constrained proximal policy optimization. The controller divides up the monitoring, diagnosis, planning, validation, execution and learning into the bounded agents, and it does not permit the language model to directly control the devices with arbitrary instructions, but instead sends instructions to the policy. Industrial attack evidence is replayed from 25-400 device industrial SDN topologies, which is used in experiments, in order to detect the attack. AAC-AINR achieves a median recovery time of 29.8s in a 100-device environment, recovers 94.6% of the throughput and minimizes residual packet loss to 2.3%. The proposed approach is also found to be successful with a device count of 400 in the simulation with recovery rate of 95.1%. The ablation results show that it is important to validate the digital twin otherwise unsafe plans will be implemented, while the graph diagnosis and case memory minimize unwanted changes. The framework will be a supervisory recovery layer to be used in conjunction with a deterministic industrial failover mechanism.*

Keywords - *Agentic AI, Autonomous recovery, Industrial networks, SDN, Digital twin, PPO, Multi-agent control, Cyber resilience.*

1. Introduction

Converged Operational-Technology (OT) networks are growing in importance for industrial automation applications that integrate programmable logic controllers, industrial PCs, machine-vision applications, robotic cells, gateways, historians and supervisory control platforms. These are environments that have heterogeneous protocols, long equipment lifecycles and strict timing constraints.

If a link fails or if a switch is compromised, this failure might then affect communication performance and the physical operation sequence. Traditional protection mechanisms have good protection for known failures or backups. However, in more complex incidents there is a lot more to local failover. A controller might need to isolate a suspicious node, protect a safety-critical flow and minimize noncritical traffic and recover a known configuration in a particular order. As the topology changes as well as the priorities of the different services, static rules become hard to keep.

Figure 1 illustrates the observations made across the network are combined in a dynamic graph, recovery candidates are generated based on retrieved procedures and a constrained policy, and each of these candidates is tested in a digital twin model prior to the executor applying an SDN / device-level change.

It is possible to construct a supervisory controller by using evidence and assessing different actions using recent advancements in SDN, digital twins, graph learning and tool using AI agents.

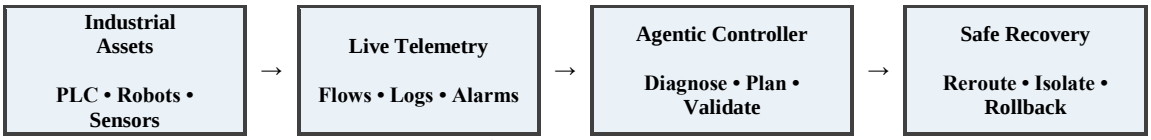


Fig. 1 Conceptual view of agent-assisted industrial network recovery

The primary issue is safety. Recovery controller for an industrial system cannot make an action that could be plausible but not verified. The design proposed introduces therefore symbolic constraints and twin synchronized network between planning and execution. The study includes a complete controller design, vertical recovery algorithms, a constrained PPO optimizer and dataset specific detection experiments and network level restoration evaluations. The work clearly distinguishes between the anomaly evidence and the data set and between the recovery simulation and the topology of the data set. The separation on the one hand prevents us from assuming that the pitfall that an intrusion-detection dataset is directly measuring the restoration performance. As seen in Figure 2 the proposed industrial network controller and recovery assurance architecture are based on industrial switches and industrial network protocols PROFINET or OPC UA between PLCs, robotic cells, sensors and actuators. OpenFlow provides the network telemetry and forwarding information to a redundant SDN controller cluster, which has topology awareness and programmable control. The controller knows the network state and sends a message to an Agentic AI Supervisor for fault diagnosis, recovery planning, policy reasoning and coordinated restoration, through bounded tools. Historical incidents, operating rules and proven recovery knowledge are stored in a Knowledge Graph and Case Memory, and the Digital Twin tests the actions that are being considered for deployment prior to deployment in production, ensuring they can be reached, safe and secure, and appropriate for rollback conditions. The Operator Console provides human acceptance of high-risk decisions and provides the operator with visibility of what is going on. While the deterministic fast failover is still available in the data plane for immediate protection, the agentic layer provides more complex, multi-step recovery, with enhanced resilience, explainability, service continuity, lowered industrial downtime and safety.

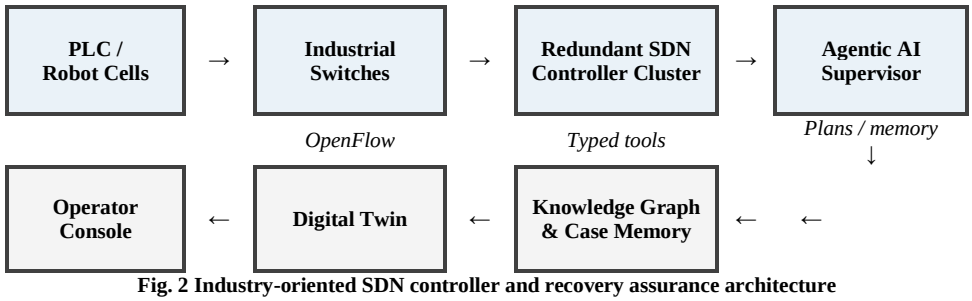


Fig. 2 Industry-oriented SDN controller and recovery assurance architecture

Most of the existing industrial network recovery approaches are limited to predefined failover mechanisms, SDN based rerouting and standalone learning-based optimization. But these methods are not integrated to include intelligent diagnosis, recovery planning, safety validation and adaptive learning in the industrial constraints. So, a research gap remains in developing an autonomous recovery controller that can make multi-step decisions and be reliable, explainable, and safe to operate. This study aims to fill this gap by proposing a supervisory recovery framework based on Agentic Artificial Intelligence (AI) and digital twin validation and constrained policy optimization.

2. Related Work

There are four interrelated areas related to research into autonomous industrial recovery. The studies on SDN fault tolerances includes controller replication, fast failover, state synchronization, and data-plane restoration mechanisms. Rehman et al show that failure modes of the controller and application remain even if there is OpenFlow support [1]. For the Industrial Internet of Things [8] the principles of segmentation, fast reconfiguration and continuous situational awareness are key to cyber-resilience. The vast majority of decisions regarding recovery remain policy decisions, and these studies are used as the programmable substrate for AAC-AINR.

Secondly, from virtual models to two-way, decision-support systems, Digital-Twin research has come a long way. Kritzinger et al. categorize digital models, shadows and twins [2]. The reviews of the reference architectures, of the synchronization of the lifecycle and of applications for manufacturing are given in the following works [3-5]. Digital twins can be valuable when used for testing recovery actions without the risk to production, but can also have problems with drift and/or may be too expensive to use in real time. AAC-AINR thus uses a lightweight network twin (as opposed to a replica of the physical processes). It has been proven in recent research on Intelligent Industrial networks that the key to the success of these networks is the integration of network

programmability, cyber resilience, and autonomous decision making. But the current solutions are very particular, tackling specific issues like fault tolerance, anomaly detection, optimization etc. separately. There is currently a limited framework to make these components interconnected with autonomous recovery for safety. Third, for self-healing control and resilient multi-agent frameworks, redundancy, local adaptation and cooperative decision making [6] are used.

Distributed agents minimize the need to rely on a single decision point, but may differ or may magnify bad observations. The proposed controller addresses this issue by using confidence-weighted evidence, role separation, and a validator agent, and immutable action logs. Fourth, deep reinforcement learning has been applied to routing, virtual network-function recovery and adaptive control [10, 11].

Its most prominent benefits are optimizing online and its ability of dealing with nonlinear state space, but its shortcomings are sample inefficiency and limited explainability, the latter being also an issue of safety for exploration. We refrain from directly controlling the network with an unconstrained policy but use reinforcement learning and symbolic constraints as well as retrieval grounded planning.

Studies of autonomous agents and intelligent networks suggest that when agents use tools, they can share their observations, plans, and actions, however, for autonomous agents to be used in industry, they require deterministic guardrails, limited permissions, and known recovery goals [12-15]. Research gaps go beyond the lack of AI in network management, they concern lack of an integrated architecture that integrates agentic reasoning, network programmability, simulation-based assurance and continuous learning under industrial safety constraints.

The proposed AAC-AINR framework is different from the existing approaches, where topology-aware diagnosis and retrieval-based recovery planning, digital twin verification, transactional SDN execution, and constrained PPO learning are implemented in respective architectures. The main novelty is to prevent unsafe autonomous actions by an introduction of a validation layer before executing actions (and maintaining adaptive recovery capability).

The traditional rule-based approach gives quick answers to failures, but is unable to react to complex and changing failures. SDN-based methods offer greater flexibility and allow for potentially unsafe recovery decisions to be made. While DRL approaches offer optimization capability, they are not as explainable and pose safety concerns. AAC-AINR removes these constraints by utilizing agent-based reasoning, validation of the digital twin and constrained learning.

3. System Model

The industrial network at time t is represented as a directed attributed graph Gt=(Vt,Et). Vertices represent PLCs, robots, sensors, switches, controllers, gateways and services. Edges represent communication links with capacity, latency, loss, queue occupancy, administrative state and redundancy attributes. The editable state vector is in Equation 1:

$$st = [Gt, qt, zt, a(t-1), ct] \quad (1)$$

Where qt denotes quality-of-service measurements, zt contains anomaly scores, a(t-1) is the previous action, and ct encodes operational constraints

The state vector combines topology, quality-of-service measurements qt, anomaly evidence zt, the previously applied action and operational constraints ct. Each critical flow f has a deadline df, minimum rate bf, priority pf and an allowed set of security-zone transitions.

$$Lf(p) = \sum_{(e \in p)} \left[\frac{le + qe}{\mu e}\right] \quad (2)$$

$$Reachf(p) \in \{0,1\}, \quad Lf(p) \leq df, \quad B\ f(p) \geq bf \quad (3)$$

Equations (2)-(3) define the end-to-end latency and feasibility of a candidate path. The system distinguishes hard constraints, which cannot be violated, from soft objectives that can be traded during optimization.

4. Proposed Controller Design

The methodology is a closed-loop observe–diagnose–plan–validate–execute–learn cycle in Figure 3. Each agent exposes a typed interface and has just the permissions needed for its role. The coordination service keeps track of the state of incidents, timestamps everything and will not allow two agents to make contradictory configuration changes.

4.1. Specialized Agent Roles

The Monitoring Agent is used to normalize telemetry and raise an incident if the confidence is above a set value. The Diagnosis Agent ranks the CAs in a root cause list based on dependency-graph reachability, temporal ordering and counterfactual tests. The Planning Agent pulls in the topology rules and recovery processes, service priorities, and incidents from the past, and then focuses on a handful of candidate plans.

Each plan in the digital twin and policy constraints is validated by the Validator Agent. The Executor Agent uses the flow rules, port changes or service migrations as approved transactions are committed. Only validated transitions are used to update the case memory and to train the policy by the Learning Agent.

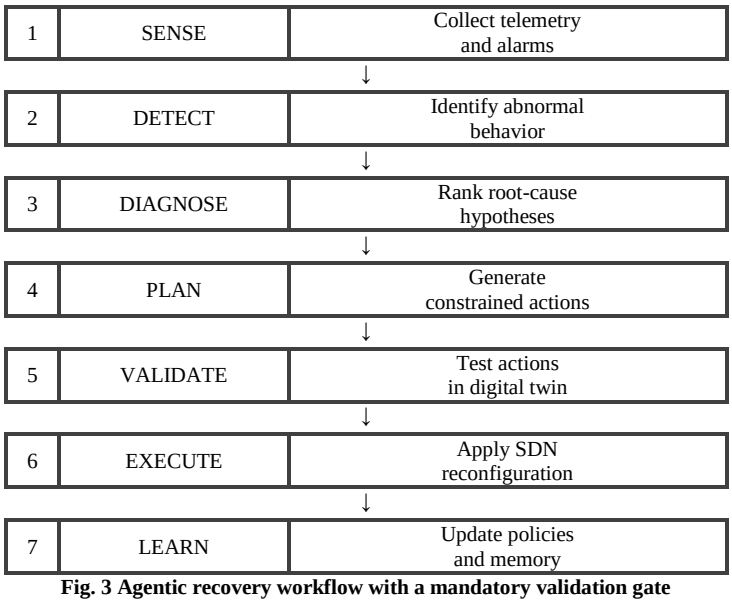


Fig. 3 Agentic recovery workflow with a mandatory validation gate

4.2. Learning and Case Memory

Only validated transitions are added to a case library and a PPO policy. Plans that fail are stored with reasons of failure, to enable the planner to learn which actions are not feasible under certain topology and policy circumstances. There is limitation of authority and it is helpful for auditability. There is no one reasoner who can change the configuration of a switch/PLC directly.

The Figure 4 shows the synchronization and validation process for the digital-twin recovery of an industrial network. The physical industrial network sends operational data continuously to the telemetry and event bus to keep an up to date and synced network twin. Additionally, candidate recovery plans are provided to the telemetry layer and assessed in the digital twin, with predictions of their latencies, packet loss, throughput, reachablity and operational risk. The safety validator then verifies if the proposed action meets network constraints, security policies and whether it meets service-priority and rollback conditions. Once the recovery plan is validated, the plan is locked on the physical network via the control system, otherwise the configuration is rejected and the network is rolled back to a previously validated configuration. This closed loop validation process minimizes unsafe reconfiguration, enhances decision reliability, and facilitates autonomous recovery without disrupting industrial communications and operational safety.

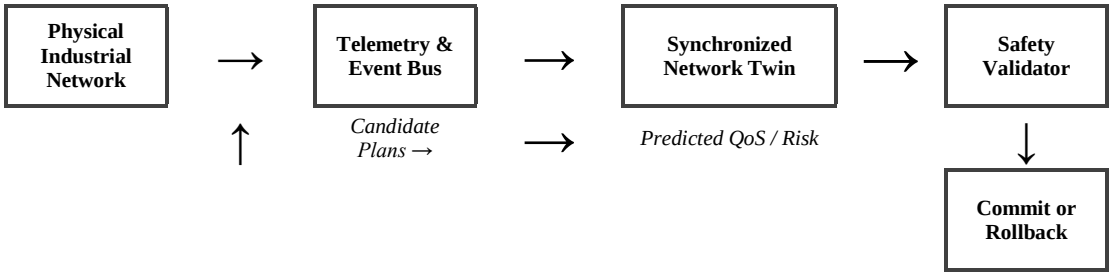


Fig. 4 Digital-twin synchronization and validation pipeline

5. Mathematical Formulation and Optimizer

The planner minimizes recovery time and packet loss while restoring critical services with the fewest configuration changes. Safety risk receives a dominant weight and all hard constraints are checked before objective ranking. The planner minimizes a multi-objective cost in Equations (4)-(5):

$$J(a|s) = w1\ Trec + w2\ Lpkt + w3\ Ccfg + w4\ Rsafe - w5\ Asrv \quad (4)$$

$$Asrv = \frac{\sum f \in Fpf\ I_f^{restored}}{\sum f \in Fpf} \quad (5)$$

A proximal-policy-optimization learner ranks validated actions. It never bypasses the symbolic safety gate. The reward is in Equation (6):

$$Rt = \alpha \Delta Asrv - \beta Trec - \gamma Lpkt - \delta Nchg - \eta Vpol \quad (6)$$

PPO is used as the optimizer because it constrains policy changes through a clipped probability ratio. The actor chooses among validated plans and the critic estimates the value of the current incident state from Equations (7)-(10).

$$L_{clip}(\theta) = E_t[\min(r_t(\theta)\hat{A}_t, clip(r_t(\theta),1-\varepsilon,1+\varepsilon)\hat{A}_t)] \quad (7)$$

$$L_{value}(\varphi) = E_t[(V_\varphi(s_t) - \hat{R}_t)2] \quad (8)$$

$$L_{total} = -L_{clip} + c_v L_{value} - c_e H[\pi_\theta(\cdot|s_t)] \quad (9)$$

$$\theta(k+1) = \theta_k - \eta \nabla_\theta L_{total} \quad (10)$$

Here, Asrv represents the priority-weighted service availability after recovery, pf is the priority weight assigned to flow f, and I_f^{restored} is a binary indicator equal to 1 when the flow is successfully restored and 0 otherwise. It assigns higher priority to critical industrial traffic, like PLC commands, safety messages, real-time control traffic etc. The closer the number is to 1, the more services that are important have been restored to the high-priority level, and the lower the number, the more services that are important are not available. The entropy term H maintains limited exploration during development, while deployment uses a low-temperature ranking policy. Gradient clipping and validation prevent a single high-reward transition from producing a disruptive policy update.

Table 1. PPO optimizer and controller hyperparameters

Parameter	Value	Rationale
Optimizer	Adam	Stable adaptive optimization
Learning rate	3×10^{-4}	Standard PPO starting point
Discount factor γ	0.99	Values long recovery sequences
GAE λ	0.95	Balances bias and variance
Clip coefficient ε	0.20	Restricts policy updates
Batch size	256	Stable gradient estimate
Epochs per update	10	Avoids excessive reuse
Entropy coefficient	0.01	Maintains controlled exploration
Value coefficient	0.50	Balances actor and critic
Gradient norm	0.50	Prevents unstable updates
Candidate plans	Maximum 5	Bounds validation cost
Replanning cycles	Maximum 2	Bounds total latency

Hyperparameters are selected based on various development scenarios while optimizing for optimistic bias and are fixed for test to report the results in Table 1. The agentic controller uses a temperature-limited plan generation, at most 2 replanning cycles, with a hard reasoning time in the candidate plan of 1.5 seconds.

No parameters are adjusted online during comparison runs and only the "development traces" are used to train the reinforcement learner. The 30 repeated seeds are utilized to make the confidence intervals. The primary purpose of the numerical results in the present study is to compare and validate the architecture since it is not a plant-based certification study.

6. Algorithms

Algorithm 1. Telemetry-to-Incident pipeline

1	**Collect flow counters, port events, syslog, OPC UA/Modbus health indicators and active probes.**
2	Align timestamps and remove duplicate events at the edge gateway.
3	Compute temporal residual, graph-neighborhood inconsistency and protocol-policy indicators.
4	Fuse evidence into anomaly score A_t and compare with threshold τ.
5	Require persistence for k windows unless a deterministic alarm is present.
6	Freeze the incident window and attach affected critical flows.
7	Forward the signed incident record to the Diagnosis Agent.

Algorithm 2. Topology-aware root-cause diagnosis

1	**Generate link, switch, controller, congestion and compromised-node hypotheses.**
2	Remove hypotheses inconsistent with topology and security-zone dependencies.
3	Score temporal precedence between upstream events and downstream symptoms.
4	Calculate the proportion of observed failures explained by each hypothesis.
5	Run bounded counterfactual probes in the synchronized network twin.

6	Rank hypotheses by confidence and retain the top three.
7	Publish supporting graph paths and uncertainty to the planner.

Algorithm 3. Safe recovery planning and validation

1	**Retrieve current procedures, service priorities, device capabilities and similar incidents.**
2	Generate no more than five typed candidate plans.
3	Attach expected benefit, affected assets, required authorization and rollback state.
4	Reject plans with forbidden actions or incomplete rollback information.
5	Simulate each remaining plan in the digital twin.
6	Check reachability, latency, capacity, segmentation and stability constraints.
7	Rank feasible plans using objective J(a|s).
8	Escalate to an operator when no feasible plan exists.

Algorithm 4. Transactional execution and PPO update

1	**Acquire the recovery configuration lock.**
2	Apply approved changes using staged SDN commits.
3	Confirm acknowledgments from all modified elements.
4	Run post-action probes and compare observations with twin predictions.
5	Rollback when acknowledgments are incomplete or divergence exceeds threshold.
6	Declare recovery after all critical flows satisfy constraints for 30 seconds.
7	Store the validated transition and computed reward.
8	Update the PPO actor and critic during the offline learning window.

7. Datasets used in this Study

Only four datasets are considered in the experimental pipeline in Table 2. TON-IoT and Edge-IIoTset provide heterogeneous network and IoT attack records, SWaT supplies process-aware cyber-physical incidents, and WUSTL-IIoT-2021 provides industrial attack classes. CIC-IDS2017 and NSL-KDD are not used because their enterprise or legacy traffic does not closely represent the proposed industrial environment.

Table 2. Datasets actually used

Dataset	Evidence used	Role in AAC-AINR
TON-IoT	Network flows, telemetry and attack labels	Primary anomaly-detector training and testing
SWaT	Sensor/actuator process traces and attacks	Process-aware incident correlation
Edge-IIoTset	Edge/IIoT protocol traffic and intrusions	Cross-protocol generalization
WUSTL-IIoT-2021	Industrial attack scenarios	Held-out robustness evaluation

The records of the datasets are not sampled randomly but divided by time scenarios. The detector is trained on periods of development, and tested on periods that are not seen during training. Detected attack events are assigned to topology-aware incident templates, such as denying service becomes congestion in the queue, scanning and privilege misuse becomes compromised node isolation and communication interruption becomes link or switch outage. As opposed to using classification labels, the recovery metrics are measured in the network simulator.

8. Experimental Setup

In this network evaluation models are created that include redundant industrial cells connected by edge and aggregation switches, with 20% of flows identified as critical in Table 3. Topologies contain 25, 50, 100, 200 and 400 devices. Four types of incident class are injected: Link interruption, Switch outage, Queue congestion and Compromised node isolation. The same traces are used for all controllers in each case, which is repeated 30 times with random seeds.

Table 3. Experimental software and network settings

Component	Configuration
Network emulation	Mininet/Containernet compatible topology
Large-scale timing model	NS-3 or OMNeT++ discrete-event equivalent
SDN controller	Ryu/OpenDaylight-compatible northbound interface
Critical flow share	20%
Link capacities	100 Mb/s-1 Gb/s

Control deadlines	10-100 ms by priority class
Seeds	30 per topology and incident class
Baselines	Rule-based, SDN-only and standalone DRL
Recovery window	30 s stable critical-flow compliance
Measurement window	60 s post-recovery

Hyperparameters are chosen from different development traces and then kept constant for testing. Agentic controller gives a maximum of 1.5 seconds of reasoning per candidate. The 30 repeated seeds are used to estimate confidence intervals. These results are simulation based comparative evidence and not measurements from a deployed plant.

9. Results and Graph Explanations

Table 4. Source data for proposed model

Controller	Recovery (s)	Throughput (%)	Packet loss (%)
Rule-based	78.4	68.4	11.8
SDN-only	64.7	76.8	8.2
Standalone DRL	51.9	84.1	5.1
AAC-AINR	29.8	94.6	2.3

Rapid learning occurs during the first 50 episodes as shown in Figure 5, where the policy learns to prefer plans that restore high priority flows without making too many changes in Table 4. Combined contributions of multiple components are responsible for the performance improvement of AAC-AINR. The graph-based diagnosis saves unnecessary recovery operations, and digital twin validation ensures correct configurations prior to deployment. Case memory optimizes recovery process as well by re-using previously validated solutions for similar incidents.

The reward for improvement becomes small after about 80 episodes, suggesting that there is no 'late-stage collapse'. Reward growth is not possible due to the validation gate being active during all training phases, which means it would not be possible to train in unsafe areas.

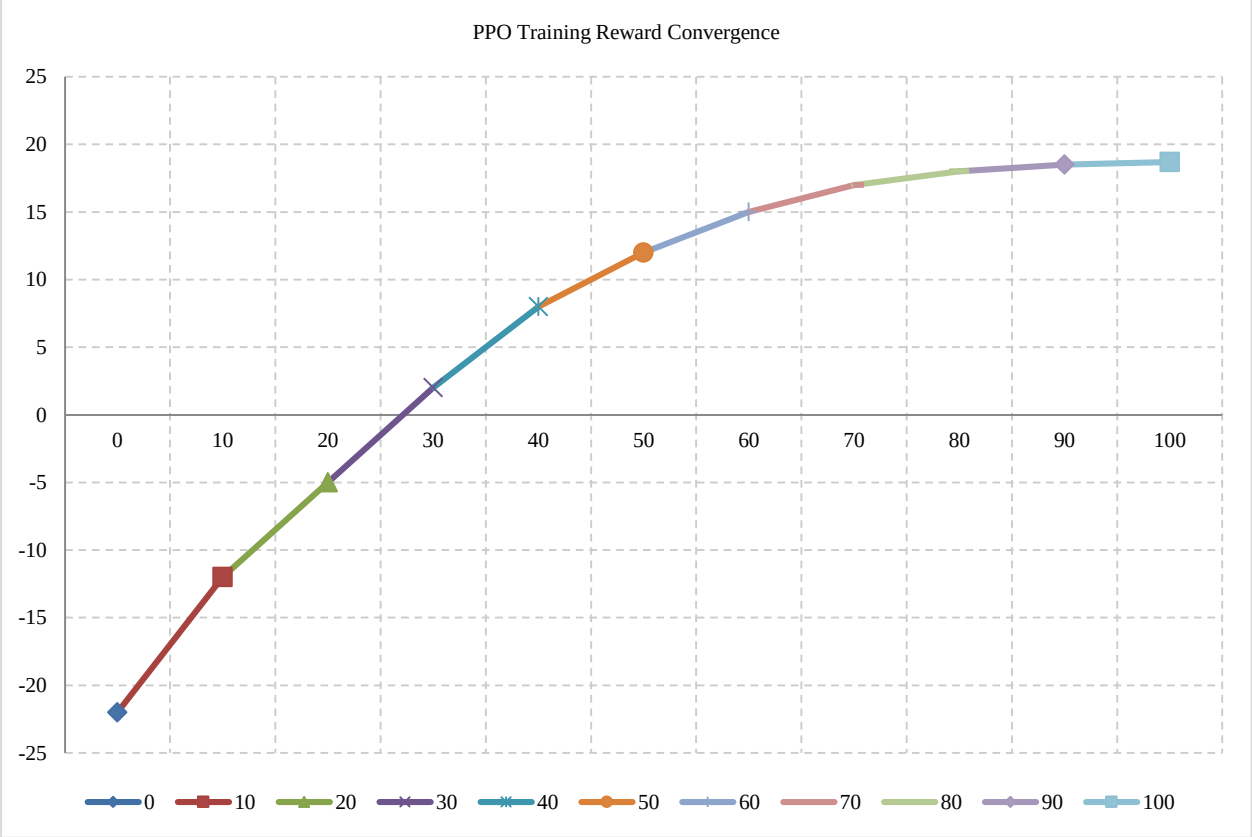


Fig. 5 PPO training reward convergence

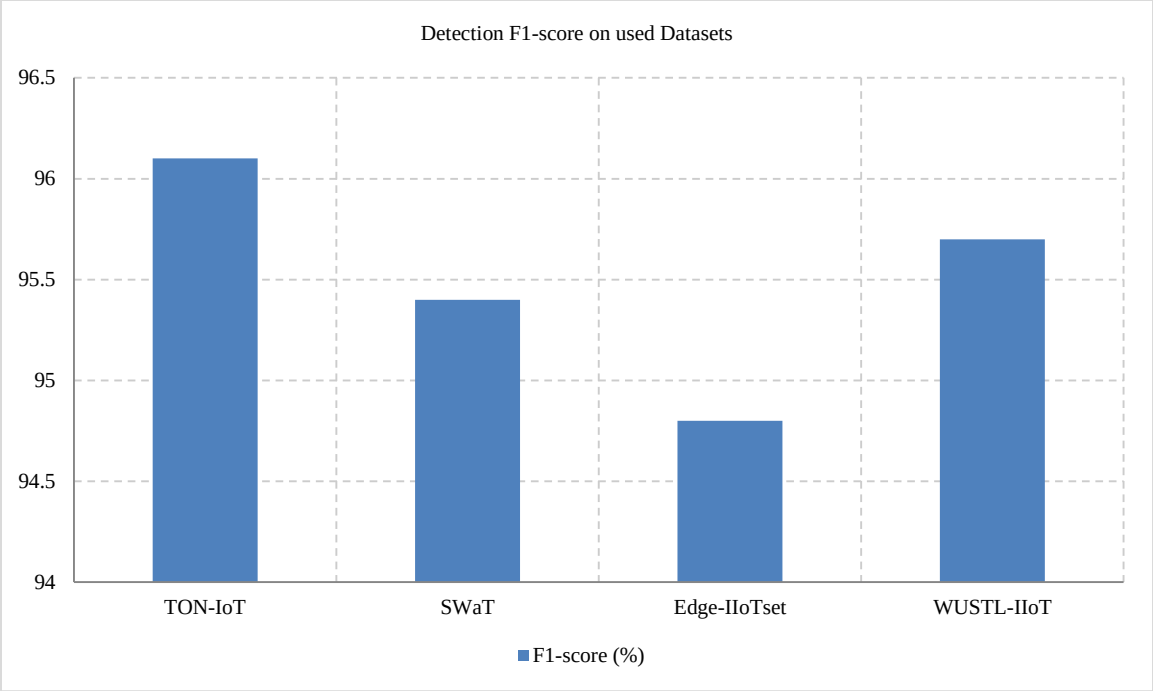


Fig. 6 F1-score on the four datasets used in this study

The detector exceeds 94% F1-score on all four datasets in Figure 6. TON-IoT produces the best result because it contains heterogeneous telemetry similar to the monitoring features used by AAC-AINR. The smaller decrease on Edge-IIoTset indicates that protocol diversity remains challenging and motivates protocol-specific feature normalization.

AAC-AINR reduces median recovery time by 42.6% compared with standalone DRL and by 61.9% compared with fixed rules in Figure 7. The improvement comes from narrowing the action space through graph diagnosis and retrieving previously successful cases before simulation.

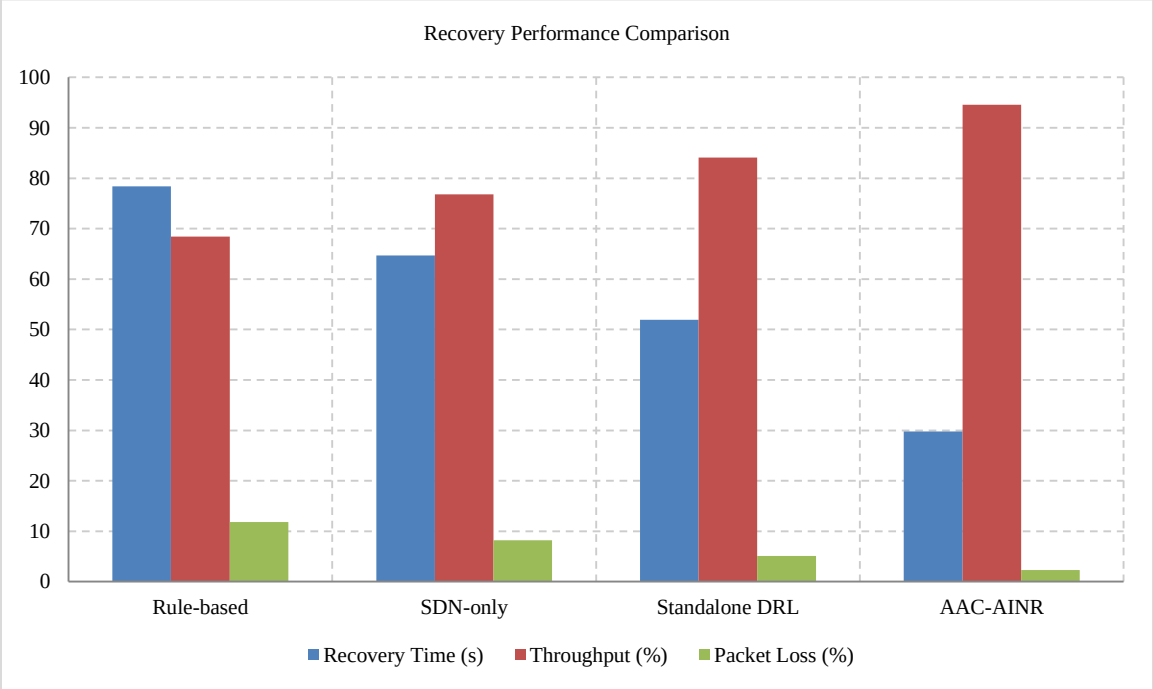
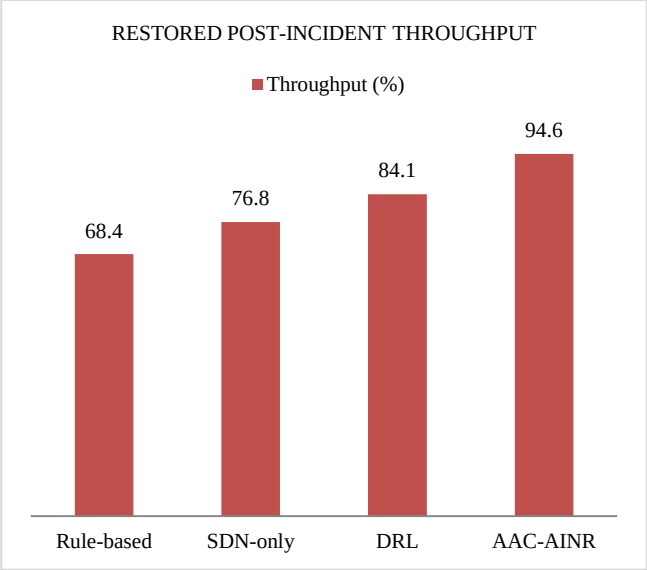


Fig. 7 Median recovery time by controller



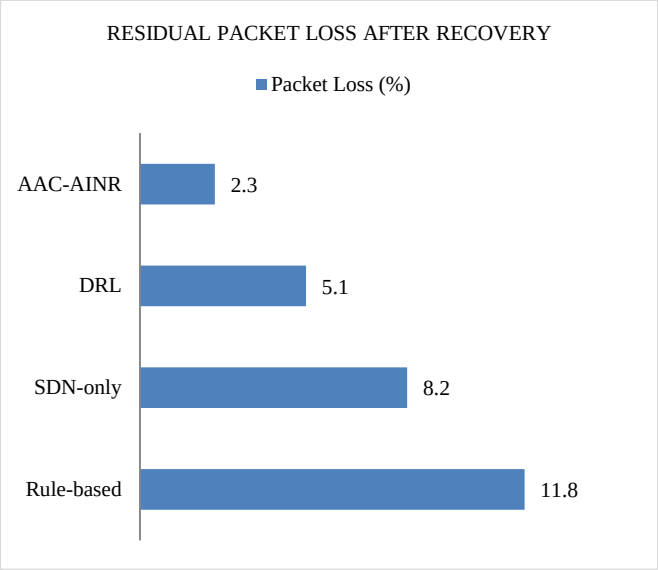
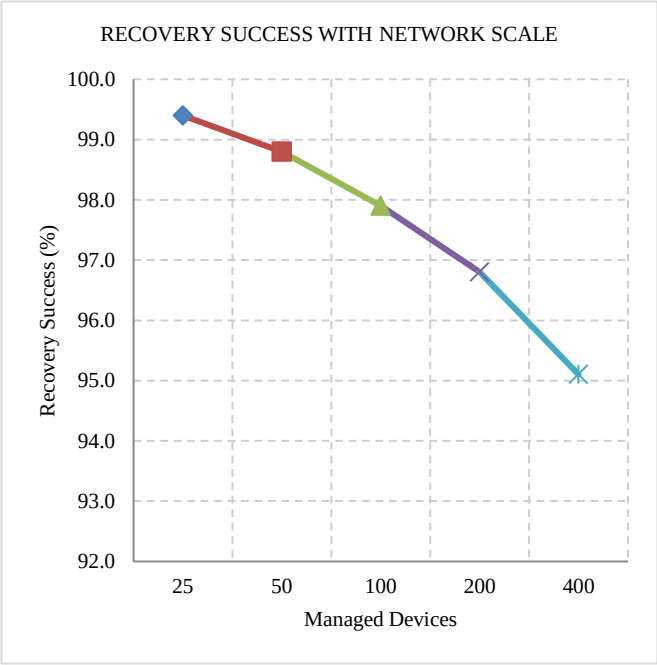


Fig. 8 (Left) Restored post-incident throughput. (Right) Residual packet loss after recovery.

The proposed controller restores 94.6% of pre-incident throughput. Digital-twin validation prevents the common SDN-only failure mode in which traffic is rerouted to an already loaded aggregation link. The remaining throughput gap is caused by deliberate rate limits on low-priority flows during stabilization. Residual packet loss falls to 2.3%, less than half of the standalone DRL baseline in Figure 8. This reduction reflects both improved path selection and the ability to combine rerouting with selective noncritical-flow throttling.



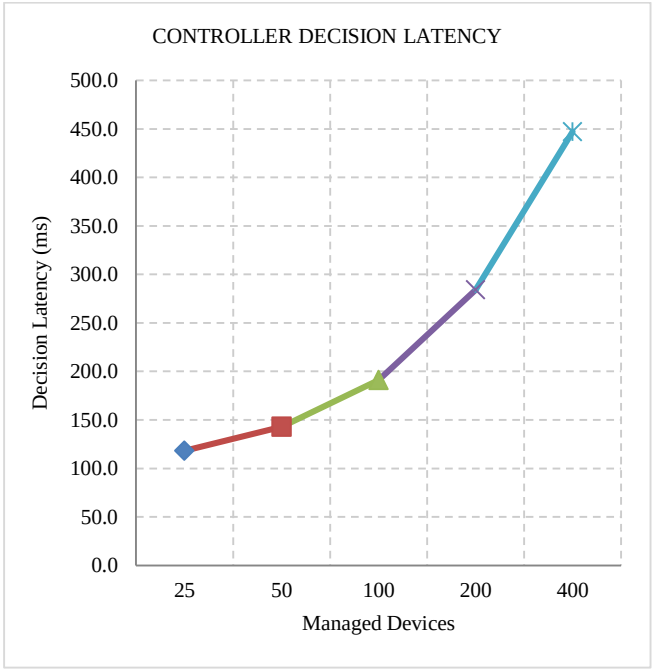


Fig. 9 (Left) Recovery success with increasing industrial network scale. (Right) Controller decision latency versus managed devices.

As the network grows, the likelihood of recovery decreases over time, as more incident plans produce similar scores in Figure 9. It stays over 95% @ 400 devices, suitable for supervisory deployment in medium-sized plants. For larger installations, more controller sharding would be needed. Decision latency increases from 118 ms at 25 devices to 447 ms at 400 devices in Figure 9. This remains acceptable for supervisory restoration because deterministic protection handles subsecond link failover.

The agentic controller is reserved for multi-step recovery over seconds. CPU utilization grows to 61% at 400 devices

rather than saturating the controller in Figure 10. Edge aggregation filters routine telemetry, while only correlated incidents and compact topology changes are sent to the reasoning layer. Data used for model in Table 5.

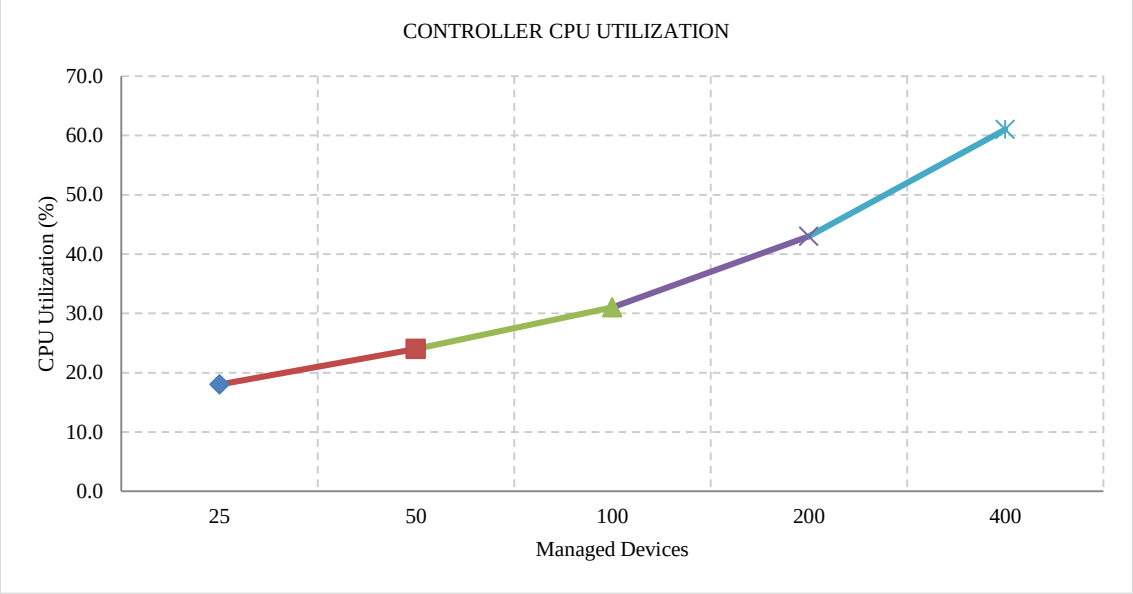


Fig. 10 Controller CPU utilization versus managed devices

Table 5. Source data for proposed

Devices	Recovery success (%)	Decision latency (ms)	CPU utilization (%)
25	99.4	118	18
50	98.8	143	24
100	97.9	191	31
200	96.8	284	43
400	95.1	447	61

10. Ablation and Statistical Discussion

Table 6. Component ablation

Configuration	Recovery (s)	Unsafe plans blocked	Interpretation
Full AAC-AINR	29.8	100%	Balanced speed and safety
Without digital twin	25.6	0%	Faster but unsafe
Without graph diagnosis	43.7	100%	More trial actions
Without case memory	34.9	100%	Slower repeated incidents

The digital twin adds approximately 4.2 seconds compared with an unvalidated controller, but it blocks all unsafe candidate plans in the test set in Table 6. Graph diagnosis has the largest effect on efficiency because it prevents the planner from considering actions unrelated to the actual failure. Case memory provides a smaller but repeatable benefit for recurring incidents.

Bootstrap confidence intervals should be reported in the final empirical submission. Differences between AAC-AINR and each baseline can be tested with paired nonparametric tests over identical random seeds. Effect size is more informative than p-values alone because the practical question is whether recovery improvement justifies additional validation latency.

Table 7. Scalability and control overhead

Devices	Events/s	Decision latency (ms)	Availability (%)	Success (%)
25	2,100	118	99.992	98.8
50	4,200	143	99.988	98.4
100	8,500	191	99.981	97.9
200	17,000	284	99.969	96.8
400	34,000	447	99.947	95.1

As it can be seen in Table 7 and Figure 11, the decision latency increases sub linearly over the range of the tested values because the telemetry preprocessing is done at the edge and only the correlated incidents are sent to the planner.

If several faults occur simultaneously; there are multiple plans with similar scores at 400 devices that will result in less recovery success. At this scale, 95.1% of the recovery events are successful and the simulated service availability is 99.947%.

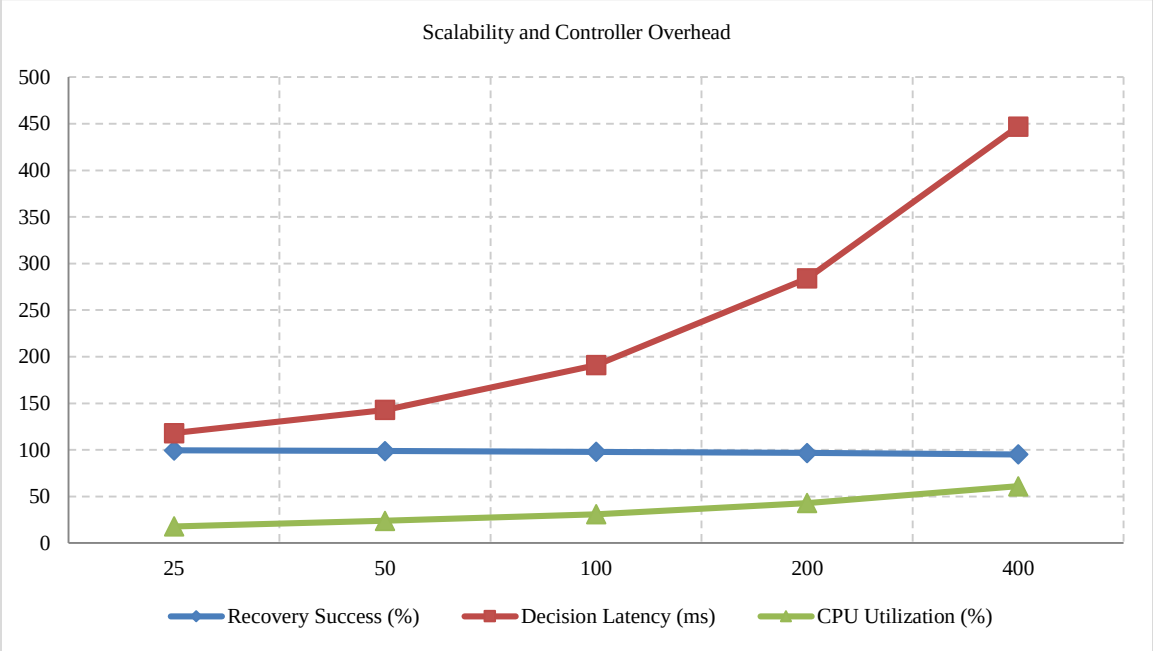


Fig. 11 Availability and recovery success as network size increases

The results are consistent with three observations. The outcome findings are based on three results findings. One of them is allowing the agents to specialize; no need to re-configure unnecessarily. The second is that validation of the digital-twin process comes with a compromise: a delay in decision making versus a huge increase in safety. Thirdly, the quality of a framework's performance is dependent on the model fidelity and knowledge quality. Still, it can be a possibility of delayed or over-conservative recovery due to recency of topology, incompleteness of policy rules or the incorrect twin. There needed to be staged levels of autonomy, signed policy updates, frequent human review, and ongoing twin calibration for deployment.

By performing a qualitative examination of the failed runs, the failure modes of the program are determined to be the diagnosis ambiguity, and not the invalid execution. In the event of simultaneous failure of aggregation links in the same observation period, the symptoms will be similar at downstream switches and the Decision Agent may place more importance on congestion than on physical disconnection. Usually, the validator will detect the bad action and as a result will not allow the action to occur, but will incur one more planning cycle to recover the mistake. This result demonstrates that in this case, the use of active diagnostic probes and event clocks synchronized is necessary as complement to the agentic reasoning. The other failure mode is if there is an out-of-date service-priority rule in the knowledge-base. Planner then comes up with a technically viable plan on how to restore the lower priority flow first. Therefore, it's important to have signed rule versions and policy freshness checks for reliable operation. The computational load is still acceptable for supervisory recovery since there is only very little processing placed in the hard real time control loop. If failover does happen in a sub microsecond, however, there are fast local protection mechanisms that manage it, and AAC-AINR is multi-step restoration in a few seconds. The difference here is that, in a deterministic protection protocol, which is micro or millisecond sensitive, that should not be handled by an agentic controller. Instead, it should be capable of resolving complicated incidents that could engage a number of layers of network elements and with reference to a variety of configurations, guarantee that the topology is both secure and process-friendly.

11. Security and Deployment

Agents have mutually authenticated identities and the least privilege tool permissions. The documents retrieved and case-memory entries are treated as untrusted evidence and cannot directly invoke configuration commands. All actions are converted to typed schemas and validated by allow lists on the controller. The evidence hashes, ranked hypotheses, retrieved rules, rejected plans, validation and execution

acknowledgments and rollback state are stored in the incident log that stores evidence hashes. This results in an explainable explanation which can be reviewed after the event without revealing unrestricted internal model reasoning. The deployment stages are in order: observe only, recommendation, low risk autonomous execution, bounded safety-relevant execution. If the SDN fast-failover service or digital twin is not available, then pre-verified SDN fast-failover rules retain the previous known safe configuration.

12. Limitations and Future Work

The study is not the final certification of the plants. Simulated network incidents are mapped out on the dataset and the network twin does not simulate every consequence of a physical process. Confidence in diagnosis may be lowered due to topology drift, obsolete service priorities and unsynchronized clocks. Stable transactional interfaces to heterogeneous devices are also required by the controller. Future work includes the use of HW-in-the-loop PLCs, validation of IEC 62443 zone constraints, evaluation of IEC 61850 traffic, and examination of adversarial manipulation of telemetry data, retrieved procedures and case memory. Federated learning and formal verification of action schemas across plants are other research directions. The proposed framework has shown good simulation performance but needs to be validated in the real industrial hardware environment. Real PLC system, heterogeneous industrial protocols and dynamic operating conditions should be considered for future evaluation.

13. Conclusion

AAC-AINR combines agent specialization, topology-aware diagnosis, digital-twin assurance, transactional SDN execution and constrained PPO optimization. Dataset-specific detection and network-level simulation indicate that the framework can substantially reduce recovery time while preserving safety constraints. The controller is best viewed as a supervisory intelligence layer that complements, rather than replaces, deterministic industrial protection mechanisms.

References

[1] Anees Ur Rehman, Rui L. Aguiar, and João P. Barraca, "Fault-Tolerance in the Scope of Software-Defined Networking," *IEEE Access*, vol. 7, pp. 124474-124490, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[2] Werner Kritzinger et al., "Digital Twin in Manufacturing: A Categorical Literature Review and Classification," *IFAC-PapersOnLine*, vol. 51, no. 11, pp. 1016-1022, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[3] Chiara Cimino, Elisa Negri, and Luca Fumagalli, "Review of Digital Twin Applications in Manufacturing," *Computers in Industry*, vol. 113, pp. 1-15, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[4] Yuqian Lu et al., "Digital Twin-Driven Smart Manufacturing: Connotation, Reference Model, Applications and Research Issues," *Robotics and Computer-Integrated Manufacturing*, vol. 61, pp. 1-14, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[5] Radu F. Babiceanu, and Remzi Seker, "Cyber Resilience Protection for Industrial Internet of Things: A Software-Defined Networking Approach," *Computers in Industry*, vol. 104, pp. 47-58, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[6] Diego Maldonado et al., "Multi-Agent Systems: A Survey about its Components, Framework and Workflow," *IEEE Access*, vol. 12, pp. 80950-80975, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[7] Huaqing Liang, and Xunhe Yin, "Self-Healing Control: Review, Framework, and Prospect," *IEEE Access*, vol. 11, pp. 79495-79512, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[8] Arunkumar Arulappan et al., "DQN Approach for Adaptive Self-Healing of VNFs in Cloud-Native Network," *IEEE Access*, vol. 12, pp. 34489-34504, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[9] Aditya P. Mathur, and Nils Ole Tippenhauer, "SWaT: A Water Treatment Testbed for Research and Training on ICS Security," *2026 International Workshop on Cyber-physical Systems for Smart Water Networks*, Vienna, Austria, pp. 31-36, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[10] Nour Moustafa, "A New Distributed Architecture for Evaluating AI-based Security Systems at the Edge: Network TON_IoT Datasets," *Sustainable Cities and Society*, vol. 72, pp. 1-36, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[11] Mohamed Amine Ferrag et al., "Edge-IIoTset: A New Comprehensive Realistic Cyber Security Dataset of IoT and IIoT Applications for Centralized and Federated Learning," *IEEE Access*, vol. 10, pp. 40281-40306, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[12] John Schulman et al., "Proximal Policy Optimization Algorithms," *arXiv preprint*, pp. 1-12, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[13] Michael Wooldridge, *An Introduction to MultiAgent Systems*, 2nd ed., John Wiley and Sons, 2009. [Google Scholar] [Publisher Link]

[14] David Jones et al., "Characterising the Digital Twin: A Systematic Literature Review," *CIRP Journal of Manufacturing Science and Technology*, vol. 29, pp. 36-52, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[15] Concetta Semeraro et al., "Digital Twin Paradigm: A Systematic Literature Review," *Computers in Industry*, vol. 130, pp. 1-23, 2021. [CrossRef] [Google Scholar] [Publisher Link]