EDITOR: Schahram Dustdar, dustdar@dsg.tuwien.ac.at

DEPARTMENT: INTERNET OF THINGS, PEOPLE, AND PROCESSES

The Troubleshooter That Never Sleeps: Continually Learning Agentic AI for Fault Debugging in IEC Continuum

Ruitao Xue, Rui Sun, Sultan Altarrazi, Devki Nandan Jha, Yinhao Li, *Newcastle University, Newcastle Upon Tyne, U.K.*

Paul Wealls, *Lenovo, U.K.*

Tomasz Szydlo, *Newcastle University, Newcastle Upon Tyne, U.K.*

Schahram Dustdar, *ICREA, UPF Barcelona, Spain*

Rajiv Ranjan, *Newcastle University, Newcastle Upon Tyne, U.K.*

The growing ubiquity of cyberphysical systems (CPSs) embedded in the Internet of Things (IoT)–edge–cloud (IEC) continuum is transforming how data-driven applications are deployed and operated. Existing approaches to fault detection, diagnosis, and healing in such CPS deployments predominantly rely on centralized or statically supervised machine learning models. Due to the growing complexity of CPS systems, such as autonomous vehicles and smart cities, which require time-sensitive responses and utilize resource-constrained IoT and edge devices, fault detection presents several formidable research challenges. In this regard, multiagentic artificial intelligence, coupled with lifelong learning, offers a promising foundation. Despite its promise, realizing such decentralized and intelligent fault-management paradigms becomes not just beneficial, but necessary.

In an era where digital intelligence intertwines with the physical world, cyberphysical systems (CPSs) have emerged as a cornerstone of modern intelligent infrastructure. CPSs integrate physical processes with distributed cyber intelligence to enable real-time sensing, decision-making, and control through tightly coupled feedback loops between computation and the physical environment.[8] Within the Internet of Things (IoT)–edge–cloud (IEC) continuum, CPSs are realized as closed-loop systems in which sensing, computation, and actuation are coordinated across heterogeneous, geographically distributed layers.[1] Resource-constrained IoT devices capture diverse physical phenomena and generate continuous, multimodal data streams under strict latency, energy, and connectivity constraints, which are processed hierarchically along the continuum. Time-critical functions, such as local filtering, inference, and immediate control, are executed close to the physical world at the IoT and edge layers, while cloud platforms aggregate system-wide data to support large-scale analytics, predictive modeling, and long-term optimization. Updated models and control policies are then disseminated back toward the edge, enabling adaptive behavior while preserving responsiveness and scalability. This architectural paradigm is exemplified by smart traffic management systems (Figure 1), where heterogeneous roadside sensors and connected vehicles feed latency-sensitive edge analytics for real-time signal control, while cloud-level intelligence coordinates city-scale optimization and learning. Across such deployments, CPS intelligence emerges from the seamless interplay

Digital Object Identifier 10.1109/MIC.2026.3662039
Date of current version 15 July 2026.

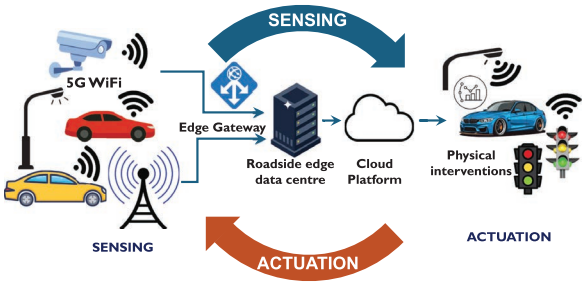

FIGURE 1. Smart traffic management system.

of distributed sensing, layered computation, and coordinated actuation, translating cyber decisions into continuous, system-wide physical interventions.

CPSs IN THE IEC CONTINUUM AND FAULT DYNAMICS

However, the same characteristics that enable CPS intelligence and autonomy also introduce substantial complexity. CPS components interact over volatile, bandwidth-constrained networks and rely on diverse hardware platforms, firmware versions, operating systems, and middleware stacks. Combined with the inherent dynamism of IoT ecosystems, such as device mobility, fluctuating power availability, and variable network conditions, this multilayered heterogeneity makes CPSs highly sensitive to run-time disturbances. This sensitivity exposes fundamental challenges in maintaining dependable operation across tightly coupled cyber and physical layers.

Against this background, the remainder of this article is organized as follows. We begin by examining the fault debugging landscape in CPSs, categorizing prevalent failure modes and analyzing how local disturbances propagate across the IEC continuum. This analysis contrasts traditional, centrally managed approaches with a decentralized, agentic paradigm for fault management. We then introduce a lifelong agent-based workflow for coordinated and collaborative fault debugging, describing how specialized agents interact to detect, diagnose, and mitigate disruptions. Subsequently, we review the enabling multiagent frameworks and algorithms, highlight open research challenges, including knowledge consistency and security, and conclude the article.

FAULT DEBUGGING LANDSCAPE

In CPSs operating across the IEC continuum, operation depends on the precise coordination of sensing, computation, and actuation across multiple layers. Disruptions at any point in this tightly coupled pipeline can therefore have far-reaching consequences. A local anomaly, such as a sensor failure or an overloaded edge node delaying computation offloading, can cascade into application-level violations of latency, accuracy, or throughput constraints. Because sensing, decision-making, and actuation are tightly coupled through feedback loops, fault diagnosis and recovery are no longer localized tasks, which require cross-layer, context-aware coordination in near real time.

To reason systematically about such failures, we adopt the Internet of Data Faults Taxonomy (IoDFT),[2] which categorizes faults along six dimensions: component, source, cause, duration, type, and pitfall. This taxonomy provides a structured framework for identifying, labeling, and analyzing faults across heterogeneous CPS environments. Furthermore, faults can be broadly classified as *known* or *unknown*. Known faults, such as a sensor producing a stuck-at value due to hardware wear, can be detected using predefined rules or models. In contrast, unknown faults emerge from unforeseen interactions, environmental changes, or evolving system behavior, and pose a far greater challenge, as they require systems capable of adaptive learning and reasoning under uncertainty.

Faults may also be characterized by their propagation behavior. *Isolated faults*, such as a transient network outage or a container crash at an edge node, typically affect a limited scope and can often be mitigated through redundancy, failover, or conventional monitoring mechanisms.[9] In contrast, *cascading faults* arise when local disturbances propagate across the interconnected layers of the IEC continuum, progressively amplifying their impact. For example, in smart traffic management systems, inaccurate data from a degraded roadside sensor can mislead edge-level congestion detection, disrupt coordinated signal control across intersections, and ultimately corrupt cloud-level predictive models. Through these spatial and temporal feedback loops, a fault originating at a single endpoint can degrade global system behavior and compromise safety or quality of service (QoS) guarantees.[10]

Traditional fault detection, diagnosis, and healing in CPS deployments across the IEC continuum predominantly rely on centralized or statically supervised machine learning (ML) models. These models are typically trained offline using historical data and deployed in a central node (often the cloud) for global decision-making. While this approach may work in static or well-characterized environments, it becomes fundamentally inadequate for dynamic, decentralized IEC settings. First, routing all monitoring data to a central authority introduces unacceptable latency and

bandwidth overhead, often violating the strict timing constraints of safety-critical applications. Second, these models assume stationary data distributions and fixed topologies, which does not hold in real-world CPS deployments where sensor behavior, network conditions, device mobility, and operational context changes constantly. Third, supervised approaches are inherently fragile when confronted with novel, rare, or compound faults that were absent from training datasets, leaving systems vulnerable precisely when resilience is most needed.

Overcoming these limitations requires a paradigm shift, from *centralized and supervised* fault management strategies to *decentralized, autonomous, and adaptive approach*. This is precisely where *agentic artificial intelligence (AI)*, combined with *life-long learning*, emerges as a transformative foundation. By deploying intelligent, goal-driven agents throughout the IEC continuum, each endowed with local perception, reasoning, and learning capabilities, fault management can become proactive, fine-grained, and inherently distributed. Such agents can detect anomalies in real time at the source, collaborate across layers to trace root causes, coordinate recovery actions, and incrementally refine their diagnostic and healing strategies without catastrophic forgetting or centralized bottlenecks. This approach not only enables faster and more granular response to anomalies, but also supports coordinated adaptation across interdependent system components, thereby preserving application-level QoS guarantees.

AI-AGENTS FOR COORDINATED AND COLLABORATIVE FAULT DEBUGGING

Building on the shift toward decentralized, adaptive fault management system, the realization of agentic AI in CPSs requires operationalizing intelligence across the IEC continuum. Rather than relying on a monolithic controller or human-driven, sequential workflows, we advocate organizing fault handling capabilities into a set of lightweight, specialized agents powered by large language models (LLMs) and equipped with tools and application programming interface (API)-calling abilities. The framework comprises focused agents, including a data *analyst/observer* that monitors streams and surfaces salient anomalies; a *diagnoser* that synthesizes causal hypotheses and estimates the scope of impact; a *planner/solutions* agent that produces concrete, policy-compliant remediations tailored to the target platform; and an *operator/executor* that applies changes safely across heterogeneous interfaces and verifies outcomes (see Figure 2). With appropriate runtime safeguards, encoded safety policies, staged rollouts, and post-action verification, these agents autonomously handle the high-volume, routine fault class while escalating only ambiguous edge cases. As a result, the CPS becomes able to explain what went wrong, suggest and apply corrective actions, and learn from each event, turning incident response from one-off troubleshooting into a consistent, time-bounded process.

The overall agentic AI approach to cascading fault debugging can be conceptualized as a three-stage collaborative process, as shown in Figure 4. Let's consider smart traffic management system where an inductive loop detector (device) reports lane occupancy to a roadside unit (edge), which in turn feeds a cloud service that maintains prediction models. Suppose the detector experiences a *stuck-at* fault (Fault X) (Step 1), as presented in Figure 3, repeatedly transmitting a constant value despite real traffic variation. *Agent A* flags the local issue through simple data quality checks (e.g., low variance) and performs a dependency analysis over the immediate pipeline. Thus, the faulty stream is consumed by edge congestion estimation and queue length inference, and it influences actuation decisions. From this dependency structure, *Agent A* predicts likely secondary effects, including (Fault Y) as corrupted or misaligned downstream features at the edge (e.g., inconsistent density estimates relative to adjacent sensors) and (Fault Z) as service-level degradation at higher layers (e.g., cloud model drift).

Agent A (Step 2) transmits a compact *incident card* (symptoms, time window, confidence, and so on) to *Agent B* (edge) and *Agent C* (cloud) exchanging insights and jointly assess the broader system-wide impact of these interconnected faults. *Agent B* validates the propagation locally by correlating neighboring sensors or cameras and checking whether the edge estimator residuals and queue estimates deviate in the affected junction. At the same time, *Agent C* assesses whether similar patterns occur across sites

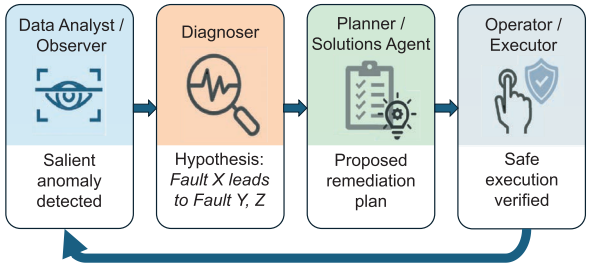

FIGURE 2. AI Agent as a composition of subexperts.

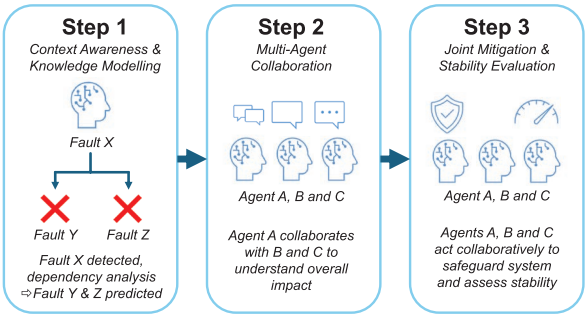

FIGURE 3. Collaborative faults debugging.

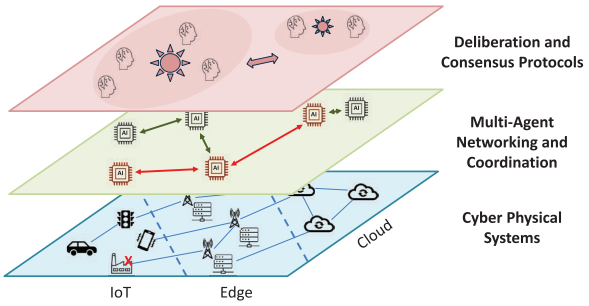

FIGURE 4. Layered architecture of the agentic AI in IEC continuum.

and whether recent model updates were influenced by the suspect feature distribution. The agents exchange evidence of the impact boundary (affected junctions, models, and prediction services), thereby establishing a consolidated and system-wide characterization of the incident and its operational scope.

Finally, the agents work together to mitigate the effects of the detected failures, coordinating protective actions and evaluating the resulting system stability in real time, thereby ensuring resilient and adaptive fault management across the IEC continuum. As coordinated mitigation (Step 3), *Agent B* quarantines the faulty stream and replaces it with a fallback estimate from adjacent sensors. *Agent C* might prevent fleet-wide propagation by excluding the identified interval from training or updates. If drift is detected, *Agent C* switches to a fallback model until validated data are restored.

AI-AGENTS DEPLOYMENT IN THE IEC CONTINUUM

The effective deployment of agentic AI relies on a hierarchical, resource-aware architecture that balances local responsiveness with global intelligence. Unlike traditional centralized monitoring, where raw telemetry is backhauled to a remote server, our approach embeds agentic capabilities directly into the infrastructure. This distributed deployment ensures that fault detection and healing occur as close to the source as possible. At the system level, we propose deploying these tailored multiagent framework across devices and sites in a way that matches available resources (Figure 4).

IoT Layer

At the foundational level, lightweight agents are embedded directly into the fabric of the IoT ecosystem, including environmental sensors, smart meters, and actuators. Due to strict power and memory limitations (often less than 256 kB random-access memory), these agents operate using quantized, small-scale inference models (e.g., TinyML). Their primary role is *observation and immediate reaction*. They monitor local data streams for gross anomalies, such as a sudden voltage spike or a frozen sensor value, and trigger preapproved, safe-mode actuations. This ensures millisecond-level protection for physical hardware without waiting for network roundtrips to the edge or cloud.

Edge Layer

The edge layer, consisting of gateways, roadside units, and 5G base stations, hosting agents with the *diagnoser* and *planner* roles. Possessing moderate computational resources (GPUs/NPUs), these agents run more sophisticated small language models or distilled versions of larger models. They serve two critical functions: 1) local correlation by aggregating data from multiple downstream IoT agents to distinguish between a single device failure and a localized system event (e.g., a broken camera versus a traffic jam), and 2) continuous fine-tuning by leveraging the lifelong learning paradigm, these agents update their internal decision policies based on immediate feedback from recent faults, adapting to dynamic topology changes without requiring a full system redesign.

Cloud Layer

The cloud layer hosts agents powered by LLMs). These agents focus on long-term memory and generalization, analyzing historical incident logs from the entire multihierarchical deployment to identify rare cascading faults that edge agents might miss. Once a new fault pattern is learned and a mitigation strategy is validated, the cloud agents compress this knowledge into lightweight policy updates, which are then dispatched to the edge and IoT layers. This creates a bidirectional flow, where telemetry moves up for deep analysis,

while intelligence and updated policies flow down for real-time execution.

Cross-Layer Collaboration

Despite this heterogeneity, LLM-based agents provide a shared communication layer using natural language and structured messages, enabling devices to exchange knowledge easily. A fault that is new for one device may be well understood elsewhere, so agents package summaries, suspected causes, reproduction steps, and proven fixes into portable *incident cards* that other agents can reuse. When incidents affect multiple subsystems or locations, agents cooperate through plain-language messages to draft response steps, including scripts, configuration changes, or rollback plans, while each site's local safeguards verify and apply the proposed actions. To respect bandwidth and privacy, exchanges prioritize short summaries and reusable response templates. Edge and cloud layers maintain a shared incident knowledge base and distribute improved response strategies back to the framework.

LIFELONG LEARNING IN THE AGENTIC AI CONTINUUM

We propose a vision of a *lifelong multiagent AI system* to overcome the inherent issues of static fault management in the dynamic IEC system, enabling detection of novel or emergent faults, adapting to evolving system topologies, or cope with environmental and workload drift. We envision our system as a decentralized set of autonomous, intelligent agents strategically distributed across the entire computing continuum. Unlike static, preprogrammed entities, these agents are designed to learn and adapt throughout their operational lifetime. Lightweight agents on IoT devices can learn to identify novel sensor-level anomalies. In contrast, more powerful agents on edge nodes can adapt their collaborative strategies to handle fluctuating network conditions or new device topologies. By continuously learning from operational data, the outcomes of past fault mitigations, and interactions with their peers, these agents collectively enhance the system's resilience. This paradigm marks a fundamental shift away from the *train-once, deploy-forever* model, creating a self-healing infrastructure that grows more robust and intelligent with every fault it encounters, without requiring costly offline retraining or manual reconfiguration.

Lifelong learning is not achieved through a single, massive training phase, but through a continuous, closed-loop process of in situ adaptation and improvement. This is fundamentally an experience-driven process. After every fault detection and mitigation attempt, each agent receives direct feedback on the outcome whether a fault was successfully resolved, if a mitigation strategy introduced new latency, or if a diagnostic was accurate. Using techniques grounded in reinforcement learning and online learning, agents autonomously update their internal models. For example, an agent might refine its dependency graph better to predict the blast radius of a sensor failure, update its communication protocol to more efficiently collaborate with a newly added edge server, or learn to discard a previously reliable but now failing mitigation playbook. The new knowledge can be shared across the agents collective, allowing the entire system to evolve its resilience and efficiency in response to the real-world challenges it faces.

Through this continuous, experience-driven process, the proposed agentic AI system becomes more adaptive, more accurate, and better aligned with real-world operating conditions. Over time, the system evolves into a robust, self-healing infrastructure capable of responding to emerging challenges without human intervention or disruptive retraining cycles.

MULTIAGENT FRAMEWORKS AND ALGORITHMS: CURRENT STATE OF THE ART

Research on agentic AI has produced a mature ecosystem of frameworks that formalize how agents assume roles, coordinate tasks, and reason collaboratively. While these tools were primarily developed for software engineering and conversational tasks, they provide the necessary structural primitives for the IEC fault debugging architecture proposed in this article.

Early systems focus on role allocation and workflow coordination. MetaGPT[a,4] encodes standard operating procedures as structured prompt programs so that role-specialized agents, such as a product manager, architect, and engineer, can carry out design, implementation, and review with clearer intermediate checks and fewer ad hoc prompts. CAMEL[b,5] introduces structured role-playing using inception prompting to guide paired agents through consistent, goal-directed dialogues, enabling controlled studies of cooperation. ChatDev[c,12] operationalizes a virtual software organization in which communicative agents follow a chat-centric pipeline with stages spanning

[a]https://docs.deepwisdom.ai/
[b]https://www.camel-ai.org/
[c]https://chatdev.ai/

requirements, coding, and testing. AutoGen[d,7] generalizes these ideas as conversation programming: Developers compose task-oriented teams (LLMs, tools, and humans) and group-chat patterns to build multiagent applications without a fixed pipeline. These designs map naturally to structured fault-debugging workflows, yet IEC prototyping introduces additional system constraints. Agents must run across IoT/edge/cloud runtimes, interact with monitoring and control stacks, and maintain bounded-latency coordination under partial connectivity.

Networking and Heterogeneity

Recent work in agent networking offers solutions to these physical constraints. Agora[11] proposes a versatile meta-protocol that balances efficiency and portability, a requirement for transmitting our proposed *Incident Cards* without saturating narrowband links. By standardizing frequent exchanges while reserving natural language for rare ambiguous faults, Agora supports the bandwidth-aware communication our architecture demands. Furthermore, AgentNet[13] and IoA[14] introduce Internet-inspired protocols for decentralized discovery and dynamic team formation. These layers enable agents to locate peers and form debugging clusters across administrative domains without routing all traffic through a central orchestrator, thereby preserving the low-latency requirements of safety-critical CPS applications.

Deliberation and Consensus Protocols

Finally, the reliability of autonomous remediation depends on robust decision-making. Debate-style protocols have demonstrated that agents can improve factual accuracy and reasoning by challenging each other's hypotheses over multiple rounds.[3] In the context of IEC fault debugging, such mechanisms allow the system to synthesize conflicting data, for instance, when an edge model contradicts cloud telemetry. However, these deliberation loops must be carefully reconciled with real-time constraints. Future implementations must employ *bounded deliberation*, where the depth of debate is dynamically throttled by the urgency of the required physical intervention.

Together, these frameworks and protocols provide the building blocks for multiagent systems that can assign roles, coordinate decisions, and run across heterogeneous devices and environments.

[d]https://www.microsoft.com/enus/research/project/autogen/

RESEARCH CHALLENGES

While the proposed lifelong agentic AI approach offers a robust framework for managing cascading faults, deploying such a system across vast and heterogeneous IEC environments presents significant scalability challenges and opens up several avenues for future research. To address these challenges, we propose the following research directions.

Knowledge Propagation and Consistency

In a lifelong learning scenario, ensuring that insights and updated models propagate efficiently and consistently across the agent population is a key challenge. Delays in disseminating critical knowledge, such as the discovery of a new fault signature, could leave portions of the system vulnerable. Furthermore, maintaining a consistent global understanding from decentralized, locally learned knowledge is an open problem.

Ensuring coherent and conflict-free actions among a large population of autonomous agents is nontrivial. Without effective coordination mechanisms, agents might take redundant or even contradictory actions, potentially leading to system instability. The complexity of reaching consensus and executing joint plans grows exponentially with the number of participating agents.

Hierarchical and Hybrid Agent Architectures

Future work should explore multilayered agent architectures that combine the benefits of local autonomy with global oversight. For instance, *regional coordinator* agents could be established on edge servers to manage and synthesize information from a cluster of device-level agents. This would reduce peer-to-peer communication overhead and provide a structured mechanism for escalating complex, multiregional faults.

From a system perspective, the challenge is not only how to program agent roles, but how to deploy and interconnect them across heterogeneous nodes. In practice, agents may need to execute in different environments (e.g., devices, edge gateways, cloud services), integrate with different observability/actuation APIs, and follow different trust and networking assumptions. This raises programming challenges (e.g., interface contracts, tool/API wrappers, safety guards, and cross-agent state management), deployment challenges (e.g., packaging, versioning, resource-aware placement, and updates across tiers), and networking challenges (e.g., protocol compatibility, routing, and

resilience to intermittency). A key practical consideration is whether an end-to-end system can be realized within a single framework, or whether multiple frameworks and runtimes are required because deployment targets and communication protocols differ across IEC layers.

Communication Overhead

As the number of agents grows from tens to potentially millions, the communication required for collaborative diagnosis and mitigation can become a major bottleneck. Unstructured, peer-to-peer communication can lead to network congestion and increased latency, negating the benefits of a decentralized approach. Research is needed into developing adaptive communication protocols for multiagent systems. These protocols should allow agents to dynamically adjust the verbosity and frequency of their messages based on network conditions, the urgency of the fault, and the relevance of the information to their peers. Techniques such as semantic communication, where agents exchange high-level insights rather than raw data, are needed.

Federated and Transfer Learning for Agents

To enable scalable lifelong learning without centralizing all data, future research should investigate the application of federated learning. Agents could collaboratively train shared fault-detection models on their local data, preserving privacy and reducing data transmission. Furthermore, developing transfer learning techniques for agentic systems would allow knowledge gained in one part of the network (e.g., how to handle a specific hardware failure) to be efficiently transferred and adapted by agents in different contexts.

Security and Privacy in Lifelong Multiagent Learning

Addressing the security and privacy implications of continuous learning and information sharing among autonomous agents is crucial. In a lifelong learning environment, agents constantly exchange data and models, which introduces vulnerabilities. Malicious actors could poison the shared knowledge by injecting false information, leading to system-wide failures. Furthermore, the continuous exchange of operational data could expose sensitive information about the system's behavior or underlying infrastructure. To mitigate these risks, techniques such as federated learning should be employed, allowing agents to collaboratively train models without exposing their raw data. Additionally, developing robust trust and reputation systems is essential to assess the reliability of information from different agents, thereby minimizing the impact of malicious entities. Future work should focus on creating secure and privacy-preserving protocols specifically designed for decentralized, continuous learning environments.

Collaborative Lifelong Learning for Fault Tolerance and Resilience

Multiagent systems can continuously learn to detect, diagnose, and recover from novel faults by sharing experiences and collaboratively adapting their strategies. This collective learning process is fundamental to building resilient and autonomous systems. When an agent encounters a new fault, it can share anonymized information about the fault's signature, the context, and the mitigation actions taken with its peers. This shared experience allows other agents to update their own models and strategies, enabling the system to more effectively recognize and respond to complex and previously unseen fault patterns. By pooling observations, the system can better correlate distributed symptoms to pinpoint the root cause of cascading failures. An active area of research is the development of scalable algorithms for knowledge sharing and ensuring the consistency of this shared knowledge, especially when faced with conflicting information from different agents.

CONCLUSION

In this article, we have highlighted the growing need for intelligent, adaptive, and decentralized fault management in the IEC continuum. As CPS deployments become increasingly complex, heterogeneous, and mission-critical, traditional static and centralized approaches to fault detection and recovery are no longer sufficient. We argued that multiagentic AI, coupled with lifelong learning, provides a promising foundation for building resilient, self-healing infrastructures capable of understanding system context, collaborating across layers, and continuously improving from experience. While significant research challenges remain, ranging from scalable coordination to knowledge consistency, the lifelong agentic paradigm charts a clear path toward robust autonomous fault debugging in future CPS ecosystems.

REFERENCES

1. M. Villari, M. Fazio, S. Dustdar, O. Rana, D. N. Jha, and R. Ranjan, "Osmosis: The osmotic computing platform

for microelements in the cloud, edge, and internet of things," *Computer*, vol. 52, no. 8, pp. 14–26, Aug. 2019, doi: 10.1109/MC.2018.2888767.
2. S. Altarrazi, T. Szydlo, S. Dustdar, S. Srirama, and R. Ranjan, "Addressing the faults landscape in the internet of things: Toward datacentric and system resilience," *IEEE Internet Comput.*, vol. 27, no. 6, pp. 43–51, Nov./Dec. 2023, doi: 10.1109/MIC.2023.3300508.
3. Y. Du, S. Li, A. Torralba, J. B. Tenenbaum, and I. Mordatch, "Improving factuality and reasoning in language models through multiagent debate," in *Proc. 41st Int. Conf. Mach. Learn.*, 2023, pp. 11,733–11,763.
4. S. Hong et al., "MetaGPT: Meta programming for a multi-agent collaborative framework," 2024, *arXiv:2308.00352*.
5. G. Li, H. Hammoud, H. Itani, D. Khizbullin, and B. Ghanem, "CAMEL: Communicative agents for 'mind', exploration of large language model society," in *Proc. 37th Int. Conf. Neural Inf. Process. Syst. (NIPS)*, 2023, pp. 51,991–52,008.
6. Z. Liu, Y. Zhang, P. Li, Y. Liu, and D. Yang, "A dynamic LLM-powered agent network for task-oriented agent collaboration," in *Proc. 1st Conf. Lang. Model.*, 2024. [Online]. Available: https://openreview.net/forum?id=XII0Wp1XA9
7. Q. Wu et al., "AutoGen: Enabling next-gen LLM applications via multi-agent conversations," in *Proc. 1st Conf. Lang. Model.*, 2024. [Online]. Available: https://openreview.net/forum?id=BAakY1hNKS
8. K. Zhang, Y. Shi, S. Karnouskos, T. Sauter, H. Fang, and A. W. Colombo, "Advancements in industrial cyber-physical systems: An overview and perspectives," *IEEE Trans. Ind. Informat.*, vol. 19, no. 1, pp. 716–729, Jan. 2023, doi: 10.1109/TII.2022.3199481.
9. H. Kayan, M. Nunes, O. Rana, P. Burnap, and C. Perera, "Cybersecurity of industrial cyber-physical systems: A review," *ACM Comput. Surv.*, vol. 54, no. 11s, pp. 1–35, 2022, doi: 10.1145/3510410.
10. S. He et al., "Cascading failure in cyber–physical systems: A review on failure modeling and vulnerability analysis," *IEEE Trans. Cybern.*, vol. 54, no. 12, pp. 7936–7954, Dec. 2024, doi: 10.1109/TCYB.2024.3411868.
11. S. Marro et al., "A scalable communication protocol for networks of large language models," 2024, *arXiv:2410.11905*.
12. C. Qian et al., "ChatDev: Communicative agents for software development," 2023, *arXiv:2307.07924*.
13. Y. Yang et al., "AgentNet: Decentralized evolutionary coordination for LLM-based multi-agent systems," 2025, *arXiv:2504.00587*.
14. W. Chen et al., "Internet of agents: Weaving a web of heterogeneous agents for collaborative intelligence," 2024, *arXiv:2407.07061*.

RUITAO XUE is a Ph.D. student at Newcastle University, NE1 7RU, Newcastle Upon Tyne, U.K. Contact him at r.xue5@newcastle.ac.uk.

RUI SUN is a postdoctoral research associate at Newcastle University, NE1 7RU, Newcastle Upon Tyne, U.K. Contact him at rui.sun@newcastle.ac.uk.

SULTAN ALTARRAZI is a lecturer at King Abdulaziz University, Jeddah, Saudi Arabia, and a Ph.D. candidate at Newcastle University, NE1 7RU, Newcastle Upon Tyne, U.K. Contact him at s.m.m.altarrazi2@newcastle.ac.uk.

DEVKI NANDAN JHA is a lecturer at Newcastle University, NE1 7RU, Newcastle Upon Tyne, U.K., and a visiting researcher at the University of Oxford. Contact him at dev.jha@newcastle.ac.uk.

YINHAO LI is a lecturer at Newcastle University, NE1 7RU, Newcastle Upon Tyne, U.K. Contact him at yinhao.li@newcastle.ac.uk.

PAUL WEALLS is a principal consultant at Lenovo U.K. & Ireland. Contact him at paulwealls@gmail.com.

TOMASZ SZYDLO is an associate professor at the School of Computing, Newcastle University, NE1 7RU, Newcastle Upon Tyne, U.K. Contact him at tomasz.szydlo@newcastle.ac.uk.

SCHAHRAM DUSTDAR is a full professor of computer science and heads the Research Division of Distributed Systems at Vienna University of Technology, 1040, Vienna, Austria, and UPF ICREA, 08018, Barcelona, Spain. Contact him at dustdar@dsg.tuwien.ac.at.

RAJIV RANJAN is the University Chair Professor for the Internet of Things research with the School of Computing, Newcastle University, NE1 7RU, Newcastle Upon Tyne, U.K. Contact him at rajranjan@ncl.ac.uk.